2014

End-to-End Classification Process for the Exploitation of Vibrometry Data

Ashley Nicole Smith
Wright State University

END-TO-END CLASSIFICATION PROCESS FOR THE EXPLOITATION OF VIBROMETRY DATA

A thesis submitted in partial fulfillment of the
requirements for the degree of
Master of Science in Engineering

By

ASHLEY NICOLE SMITH
B.S., Wright State University, 2012

2014
Wright State University

Wright State University
GRADUATE SCHOOL

July 28, 2014

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Ashley N. Smith ENTITLED End-to-End Classification Process for the Exploitation of Vibrometry Data BE ACCEPTED IN PARTIAL FULFILLMENTOF THE REQUIREMENTS FOR THE DEGREE OF Master of Science in Engineering.

_____
Arnab Shaw, Ph. D.
Thesis Director

Committee on Final Examination

_____
Brian Rigling, Ph. D., Chair
Department of Electrical Engineering

_____
Arnab Shaw, Ph.D.

_____
Fred Garber, Ph.D.

_____
Brian Rigling, Ph.D.

_____
Matthew Dierking, Ph.D.

_____
Robert Fyffe, Ph.D.
Vice President for Research and
Dean of the Graduate School

ABSTRACT

Smith, Ashley Nicole. M.S.Egr., Wright State University, 2014. End-to-End Classification Process for the Exploitation of Vibrometry Data.

Laser vibrometry provides a method to identify running vehicles' unique signatures using non-contact measurements. A vehicle's engine, size, materials, shape, and other attributes affect its vibration signature. To develop the capability to classify and identify these signatures, a robust aided target recognition (AiTR) end-to-end process is evaluated and expanded. The main challenge in classifying a vehicle's vibration signatures is presented by the operating conditions and parameters that vary as a function of sensor, environment, and collection locations on the target, among others. Some of the parameters affecting the vibration signatures include weather, terrain, sensor location, sensor type, and engine speed. Another challenge in vehicle classification is the determination of signal features that can overcome the differences created by these varying operating conditions. The end-to-end process consists of signal preprocessing, feature extraction, feature selection, classification, and identification. A total of 11 features from automatic speech recognition, seismology, and structural analysis and previously utilized in vibration exploration were used in this end-to-end process. Features were selected by two feature selection methods to determine the best feature set for vehicle classification. Finally, four classifiers were used to identify the vehicles'

signatures. Confusion matrices were used as metrics to evaluate the effectiveness of the end-to-end process. The entire process was tested on two sets of data: a military vehicle collection using accelerometers and a civilian vehicle collection using a laser vibrometer and accelerometers.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

ACKNOWLEDGEMENT

I would like to thank Dr. Arnab Shaw and Dr. Matthew Dierking for allowing me the opportunity to work at the AFRL Sensors Directorate. I would like to thank Dr. Shaw for advising me throughout my thesis work, particularly for all his help in understanding feature extraction, and Dr. Dierking for his guidance in all things of laser radar signal processing. I would like also like to thank Brian Stadler, Fred Heitkamp, Olga Mendoza-Schrock, and Scott Kangas for all of their assistance and guidance. I would like to thank Clare Mikula, Stu Shelly, and Darrell Barker for letting me use their vehicles for my data collection and Larry Barnes, Kevin Sigmund, Cpt. Ben Roth, 2Lt. Drew Stroshine, Randy Depoy, Bruce Smotherman, and Dirk Adams for all of their assistance with the data collection.

Lastly, I would like to thank my friends and family for all of their support, motivation, and encouragement to put my best foot forward and reach for the stars in all things I do. This effort was supported in part by the U.S. Air Force and Wright State University through contract number FA8650-12-D-1377. The views expressed in this thesis are those of the author and do not reflect on the official policy of the Air Force, Department of Defense, or the U.S. Government. This thesis was approved for public release via 88ABW-2014-4909.

1. INTRODUCTION

1.1. Problem Statement

Laser vibrometry provides a target recognition method that is independent of a vehicle's geometry (however signatures may vary with vehicle geometry). For this reason, laser vibrometry provides a novel phenomenology to exploit that enhances current vehicle classification technology. Laser vibrometry measures the micro-velocity of vibrating surfaces due to internal plant motions and records the resulting signatures as a function of time. These vibration signatures are utilized to classify vehicles for multiple applications. As an example, identification using laser vibrometry can be used for cataloging vehicles travelling to areas, which serves as an automated way to secure an installation. Another example consists of using laser vibrometry data to classify a military vehicle as friend or foe, serving to reduce friendly-fire situations. A myriad of additional examples are possible for a wide range of applications.

The exploitation of the vibration signatures proves to be an interesting challenge. Different sensors, beam locations, speed, engine activity, terrain, weather, and more affect vehicle's signature and classification results. These operating conditions are variables that can change between data collections. These varying operating conditions are what make target identification a difficult problem. The challenge is identifying a method which is robust enough to overcome these variations in the vehicles'

signatures. The development of an end-to-end process that can help develop these robust algorithms is the primary purpose of this thesis work.

An end-to-end process was developed based on the work done previously by Charles River Analytics (CRA) and the Air Force Research Laboratory (AFRL) [1, 2]. In order to have a basis for comparison, 11 features from [1] were chosen to describe the data. These features were originally chosen in the previous work based on their availability in the literature on seismic and acoustic processing. They provide a mechanism for process comparison, but do not represent an optimized set of features for remote vibrometry. Four of the selected features are time domain features and the remaining seven are frequency domain features. As some of the features create multiple results, 33 total feature values are utilized to characterize the vehicles. For example, two of the features implemented in [1], Mel-frequency cepstral coefficients (MFCC) and linear prediction coefficients (LPC), each output 10 feature values every time they are calculated. In this work, as in each of the previous efforts, the input to the features is the preprocessed accelerometer data.

Features were selected from the set of 33 to further reduce dimensionality using two feature selection filters. The selected features were used to classify the vehicles using four different classification algorithms. Finally, using a human-in-the-loop process, the end-to-end process was analyzed and fine-tuned as needed.

To again have a basis for comparison in the evaluation of this end-to-end process, the same set of contact sensor data was used as in previous work [1]. In addition, a new four target dataset was also collected to examine both remote velocity and contact acceleration measurements in the process. The end-to-end process was run on the older collection and

new data collection to examine the performance of the process. The last analysis step of this work does compare the process based on the overall success of the classification; however, since only limited datasets were used as well as features selected from different fields, only the efficacy of the end-to-end process is important here, not the specific classification performance. This end-to-end process will be utilized in future research to develop optimized laser vibrometer specific features and classification processes.

1.2. Previous Work

Several publications discuss analyzing and exploiting vibrometry data for target identification. Each publication approached the problem differently but had the goal of identifying vehicles or information about the vehicles. The previous works varied sensor phenomenologies used to collect the vehicle signatures and signal features used for classification and identification. Directional microphones were used to identify four vehicles in different traffic situations by Nooralahiyan and his coauthors [3]. The authors extracted linear prediction coefficients (LPC) and used time delay neural networks (TDNN) to classify the vehicles in a small set of traffic patterns. Ten LPC were calculated for 25 frames of data for different speeds and vehicles [3]. Using these LPC, the TDNN was able to achieve 95% convergence and 100% classification of vehicles [3]. Using LPCs and TDNN to classify vehicles was highly successful in this small case.

A paper was published a few years later by CRA and AFRL about a generic aided target recognition (AiTR) algorithm comprising of 11 features, one feature selection method (ReliefF) and four classifiers designed to identify vehicles using accelerometer data [1]. The dataset consisted of three vehicles and a power transformer. In order to calculate the features, the authors created 200 millisecond (ms) windows of data. The top

five features (selected by ReliefF) were used in four different classifiers with 74% to 98% success in classification [1]. CRA and AFRL published a second paper the following year using a new data set with a focused target identification process using only four features and a clustering classification algorithm [2]. They took 100 ms windows of data, calculated the features and used ReliefF to select features for three vehicles. The features were used to successfully determine trait similarities between the vehicles by using K-nearest neighbor with varying cluster sizes [2].

Masagutov and his coauthors, including one from CRA, developed a novel classification method for determining vehicles with unknown engine throttles by training with limited known engine throttles [4]. The authors simulated laser vibrometry measurements with accelerometers. The data was resampled to 1 kHz, transformed to the frequency domain, and normalized using Euclidean norm [4]. A singular value decomposition (SVD) was performed to create a background signature, which is stable to throttle level, and a foreground signature, which stretches and grows in magnitude for varying throttles. These signatures were recorded as vibration models for three vehicles and multiple accelerometers in a library to compare to the testing data. The testing data was compared to the foreground and background signatures and an error value is calculated between them [4]. The error value was minimized, creating target classification, by running the testing data over all available models. Using this method, the authors achieved 90%+ classification for a two class problem.

Kangas and his coauthors revived CRA's work by examining manifold learning to reduce data dimensionality for a novel approach to vehicle target recognition [5]. The authors used principle component analysis (PCA) and diffusion maps to reduce

dimensionality and classify three vehicles using four classification algorithms. The two techniques were applied to time series data and power spectral densities (PSD). High classification rates, 93% and higher for the time series data and near 100% for the PSDs, were seen using a small number of dimensions relative to the total length of data.

A different paper placed wireless accelerometers and magnetometers on the highway to classify vehicles based on the number of axles they have [6]. The authors used the data they collected to determine axle counts, arrival and departure times, speed estimates and the number of vehicles that passed over the sensors for classification. The sensors were used to detect axle spacing, speed and position/time where one vehicle begins and another ends. These factors were used to determine and sort vehicles by their axle count. A classification rate of 99% was achieved, even during high traffic conditions [6].

An exploration of the effects of assuming stability in vibration signatures over different locations on a vehicle's surface was completed by Crider and Kangas [7]. The authors used peak detection and compared the peaks of the accelerometers across the vehicles to determine if there were similarities. They truncated the bandwidths of the signals, identified peaks and peak locations, and plotted the peaks over the different accelerometer locations together. They determined most of the vehicle's energy is lower in frequency and is relatively stable across the vehicle [7].

Lastly, a Swedish report examined the performance of four different feature extraction methods on military vehicle data collected with laser vibrometers [8]. The data was collected on six vehicles from distances between 200 and 1900 meters with different engine revolutions per minute (rpm), different illumination angles, and on different parts and surfaces of the vehicles using a 1.5 μm coherent laser radar. Each set of data was

windowed to 1024 samples and features were extracted. The four feature vectors used by the author were the 6 peaks with the highest amplitudes from the PSD, the 6 modeling parameters from a 6^{th}-order autoregressive model, the 6 peaks with the highest amplitude from the marginal frequencies calculated with the Morelet wavelet, and characteristic parameters from the first 6 modes of an empirical mode decomposition (EMD) [8]. Over 200 instances of recorded data were classified using a Mahalanobis distance classifier. A 6 class comparison produced a correct classification rate between 38% and 62% for the different feature extraction methods [8]. The PSD features performed the worst while the EMD features performed the best.

Pros and cons exist in the analysis methods from each of these previous works. Using LPCs and TDNN to classify vehicles for traffic management in [3] provides highly successful classification for their small dataset. These results, however, may not transfer well to larger datasets because the selected features are not optimized for use when multiple operating conditions exist. The work done by Stevens and his coauthors laid way for an end-to-end framework to be developed [1, 2]. Their work, however, does not appear to provide large amounts of results to prove their features and classifiers successfully classify vehicles with multiple sensors and operating conditions. The novel classification method developed by [4] classifies with 90%+ classification rates for a piston vs. turbine comparison when using contact accelerometer data. The models used to classify, however, does not account for the noise in remote velocity. Without taking the noise of remote velocity their classifiers may not adapt to laser vibrometry data.

The previous work reveals a gap in the research. Much of the vibrometry target identification research has been accomplished using accelerometers as a surrogate for

remote vibrometry data. The goal of this approach was to provide high bandwidth, high diversity, and low noise signals to begin the AiTR development. Additional research is needed to allow contact measurements to accurately predict remotely collected signatures. The state-of-the-art in classification appears to be between 95% and 100% using idle data and between 74% and 100% when using multiple stationary conditions for very small numbers of targets and operating conditions [1, 5]. Data from moving targets has not yet been exploited for precise target identification. Real remote vibrometry data, not simulated, has only just begun to be exploited to classify vehicles. The work completed in this thesis combines existing techniques from multiple papers, [1], [2], and [5], to use on contact accelerometer data and remote velocity data.

1.3. Layout of Thesis

The remainder of this thesis is divided into six chapters. Chapter 2 introduces laser vibrometry and aided target recognition as background for this work. The end-to-end process and the steps within it are examined and explained in Chapter 3. Chapter 4 describes the two datasets that are used to evaluate the end-to-end process. The results from using the end-to-end process on the two datasets are detailed in Chapter 5. Finally, Chapters 6 and 7 cover the conclusions of this research and possible future work.

2. BACKGROUND

2.1. Laser Vibrometry

Laser vibrometry, also known as laser-Doppler vibrometry, measures vibrating surfaces using the Doppler effect. Laser vibrometers are non-contact sensors measuring velocity or displacement as frequency or phase shifts that occur when light is scattered from a target [9]. The size of the shift is dependent on the velocity of the target's vibrations and the wavelength of the laser vibrometer. These systems can take single point 1-D measurements or scanning measurements depending on how they are built and the type of measurement needed for fault detection, target recognition, and other applications.

Continuous wave (CW) systems and pulsed pair systems are two common implementations of laser vibrometry systems. CW laser vibrometers are frequently used within commercial and industrial applications while pulse pairs are more frequently used for long range sensing [10]. The laser vibrometer used for this research is a commercial single point CW system consisting of a Polytec OFV-503 sensor head and an OFV-5000 acquisition box. The continuous wave system will be further explained in this section. Single point CW systems precisely measure the small frequency shifts in the light that was returned from the vibrating target using an interferometer [9]. Figure 1 shows a simplified optical setup of the interferometer-based laser vibrometer.

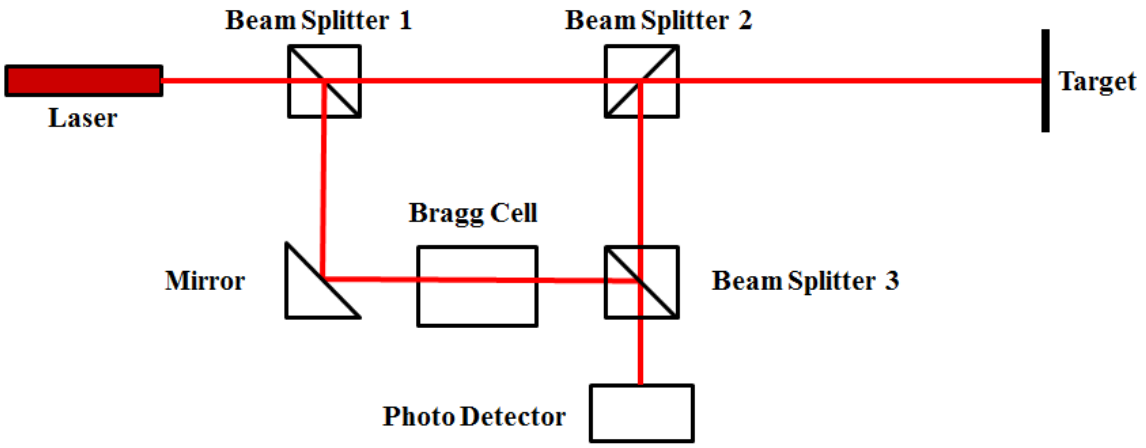


Figure 1. Optical components of a continuous-wave laser vibrometer.

The interferometer works by splitting up the helium neon laser's light using a series of optical beam splitting mirrors to create a reference beam and a measurement beam [9]. The reference beam is passed from beam splitter 1 to the mirror, into a Bragg cell and through beam splitter 3 into the photo detector. A Bragg cell is an acousto-optic modulator used to shift the reference beam in frequency [9]. This frequency shift provides a way to measure both magnitude and direction of the velocity and displacement of the target. The measurement beam goes through beam splitters 1 and 2 to the target's vibrating surface. The reflected measurement beam is passed back through beam splitter 2 and down through beam splitter 3 into the photo detector. The Doppler shift is proportional to v/λ, velocity over wavelength. At the wavelength of the Polytec, 633 nm, the frequency shifts due to typical target motions will range from fractions of a Hz to a few hundred Hz. The Doppler frequency shifted measurement beam and the frequency offset reference beam create an interference signal on the photo detector that when recorded over time is directly related to the velocity of the target's vibrations [9].

2.2. Aided Target Recognition (AiTR)

Aided Target Recognition (AiTR) algorithms and devices are used to detect and identify targets using a variety of sensors such as imaging systems, laser vibrometers, synthetic aperture radar (SAR), and 3-D ladar systems [11]. AiTR systems and aided target recognition methods find efficient ways to combine human and computer analysis [12]. Automatic target recognition (ATR) systems make decisions solely on computer analysis. ATR systems require enormous amounts of data to work efficiently, so AiTR systems are used instead. Pattern recognition and knowledge-based processing are the two main types of AiTR algorithms. AiTR algorithms using pattern recognition analyze targets based on statistical and structural analysis techniques. Knowledge-based AiTR algorithms approach detection and identification as an artificial intelligence learning from and adapting to its environment [11]. Pattern recognition algorithms were used for this research as an initial approach to AiTR.

Figure 2 illustrates the five steps typically used in pattern recognition algorithms: preprocessing; detection; segmentation; feature computation, selection, and classification; and decision [11]. Preprocessing often involves noise reduction and filtering to improve the quality of the target data. The detection step determines whether the target is located within the measured area. Segmentation extracts the suspected target from the data and uses it to compute the defining features. Features are selected to maximize differences between targets and targets are classified based on the features. Finally, a decision is made based on the classifier's output.

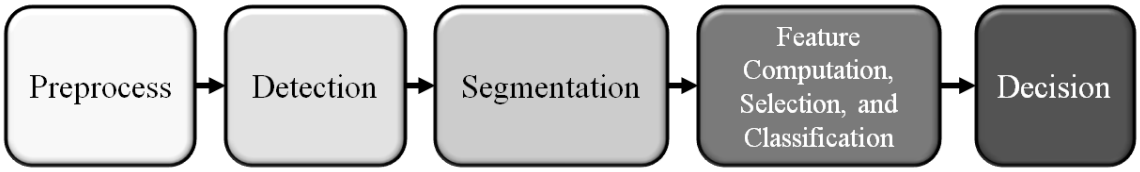


Figure 2. The general end-to-end process for AiTR.

The AiTR process is generalized to work for primary sensors that will need to detect and identify targets. Laser vibrometry works as a complementary sensor analyzing a target after it has been detected by another sensor. Since laser vibrometry is being used as a complementary system, the general AiTR process is reduced removing the second and third steps of the generalized AiTR process. The detection and segmentation steps are not used in the end-to-end process for vibrometry. Detection is not necessary because another sensor would be used to detect the target. Segmentation is also not necessary because the target would be located by processing the data collected by other sensors as well. Laser vibrometry would be used to analyze and identify the detected target.

3. THE END-TO-END PROCESS

An end-to-end process was developed to provide a framework to exploit vibrometry data for vehicle classification. The process consists of five distinct steps: (1) preprocessing data, (2) feature extraction, (3) feature selection, (4) classification, and (5) analysis of the process [13]. A simple diagram of the overall end-to-end process is shown in Figure 3. A generalized process was developed to provide an easy mechanism to add or change individual processing methods and algorithms. These individual steps are used to introduce new or additional techniques, features, or classifiers to test and improve vibrometry exploitation. Many combinations of features, feature selectors, and classifiers exist using this end-to-end process. A starting point including 11 features, two feature selectors and four classifiers that were chosen from different parts of the work done previously by Stevens, et. al and Kangas, et. al [1, 5].

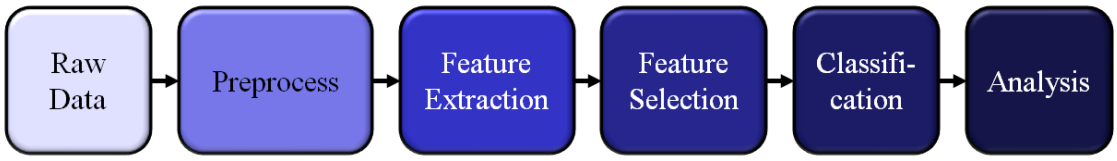


Figure 3. Diagram of the end-to-end process for vibrometry.

The end-to-end process was completed using a combination of MathWork's Matrix Laboratory (MATLAB) and the Waikato Environment for Knowledge Analysis (WEKA). WEKA was developed by the University of Waikato for data mining and

machine learning. WEKA is a toolbench used as a standalone program or may be manipulated and utilized in Java, MATLAB, and Python [13]. MATLAB code was used to preprocess the vibration data and to extract features. The standalone WEKA program was used to select features, classify the targets, and help analyze the results of the overall end-to-end process. Each step of the end-to-end process is further explained in detail within the sections of this chapter.

3.1. Preprocessing

Raw remote velocity and contact acceleration data require preprocessing before feature extraction can take place. Acceleration is the rate of change of velocity. Since the accelerometers are being used as a surrogate measurement for the remote vibrometer, the acceleration data is integrated to velocity data to provide an estimate of the vibrometer output. Integration of the acceleration will average out the overall noise. Integration on the accelerometer data is performed instead of a derivative on the laser vibrometer data because vibrations measurements from a long range are typically noisier than those closer to the target. The differentiation of the laser vibrometer data would greatly amplify the noise. Figure 4 and Figure 5 show the steps required to represent the two sensors similarly.

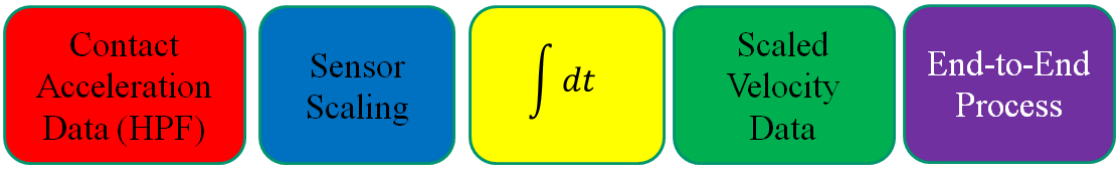


Figure 4. Steps of preprocessing for accelerometer data.

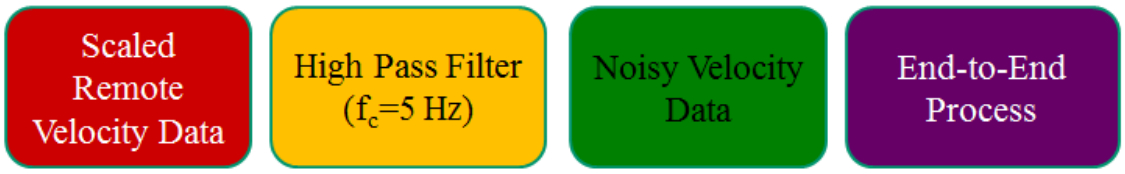


Figure 5. Steps of preprocessing for laser vibrometry.

Accelerometer data is collected with a 10 kHz sampling rate. When the data is collected, the acquistion system high pass filters (HPF) the data with a cutoff frequency of 5 Hz to remove low frequency noise. The data is converted to acceleration units (mm/s^2) and scaled using the individual sensor's sensitivity. The scaled acceleration data is integrated and passed along to the next steps for preprocessing.

Similarly, the laser vibrometer also collects data with a sampling rate of 10 kHz. The velocity data is scaled by the acquisition system before the data is saved and converted to velocity units (mm/s). To remove some of the small noise oscillations the data is high pass filtered with a cutoff frequency of 5 Hz. The filtered velocity data is passed onto the next stage of the end-to-end process for further processing.

Once the data is formatted into a velocity vector, each trial of time series data is divided into one second sections (10,000 data points each). Each section is windowed to five 2,048 point sections. Each window, except the first window, is overlapped 25% with the previous window. Each window is normalized to zero mean and unit norm. The data is made to be zero mean by subtracting the mean of each window. Unit norm is accomplished by finding the Euclidean norm, or L_2 norm, and dividing each window by the norm value. A power spectral density (PSD) is taken of each normalized window to use for part of feature extraction section. The complete preprocessing cycle is shown in Figure 6.

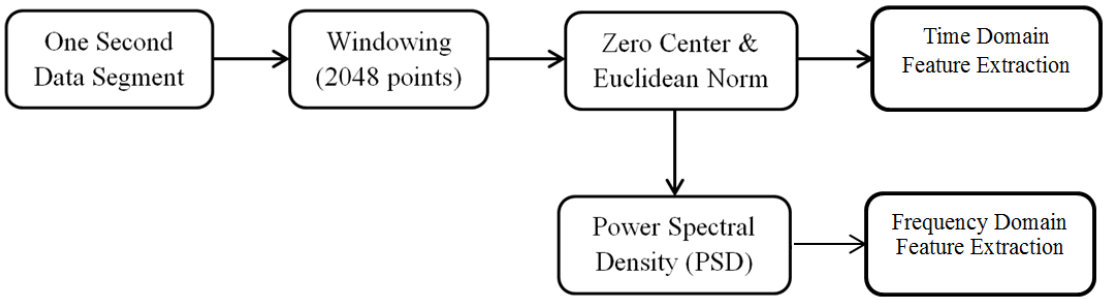


Figure 6. General preprocessing steps for the end-to-end process.

3.2. Feature Extraction

Feature extraction is a method to reduce data dimensionality [14]. Features represent relevant information in target data to distinguish between target classes. Features are not expected to fully represent the target data they are extracted from. However, the goal of feature extraction is to preserve maximum amounts of information while minimizing the overall computational complexity of the process. CRA and AFRL's original research created a list of 11 features [1]. These 11 features were used to start testing this end-to-end process, but were assumed not to be optimized for this application or remote vibrometry data. Although in many of the features there are redundancies or features based on timescales inappropriate for typical signatures, they do provide a good basis of comparison to earlier work. The features were selected from existing applications within the automatic speech recognition (ASR), seismology, and structural analysis communities [1]. Of the 11 features, four features are derived from the time domain and the remaining seven are in the frequency domain. Complexity, root mean square (RMS), reflection coefficients from linear prediction coefficients (LPC), and zero crossing rates are four time domain features used. The seven frequency-domain features are dominant

frequencies, flux, Mel-frequency cepstral coefficients (MFCC), peak counting, rolloff, spectral ratios, and spectral centroid.

All of the features are calculated the same way as CRA and AFRL calculated the features in the aforementioned studies [1]. After the target data is preprocessed, features are calculated from each window of 2,048 points. Time domain features are extracted from each of the five windows and averaged together. Frequency domain features are extracted from the PSDs of each data window and averaged. The 33 averaged features are used to represent the target data. Each feature is explained in detail within the next two subsections.

Time-domain features

Zero Crossing: Zero crossing is a sum of the number of times a zero centered signal switches from a positive point to negative point or vice versa. Zero crossing is calculated in Equation 1, where x_i and x_j are adjacent data points and N is the total number of data points. Zero crossing is not frequently used by itself but rather manipulated into the number of zero crossings within a specific period which is also known as a zero crossing rate. This manipulation is done by only finding the number of zero crossings in a very small time period.

$$\boldsymbol{ZC} = \sum_{\boldsymbol{i=0,j=1}}^{\boldsymbol{N-1}} \begin{cases} \mathbf{1} & (\boldsymbol{x_i} > \mathbf{0}) \vee (\boldsymbol{x_j} < \mathbf{0}) \\ \mathbf{1} & (\boldsymbol{x_i} < \mathbf{0}) \vee (\boldsymbol{x_j} > \mathbf{0}) \\ \mathbf{0} & \boldsymbol{otherwise} \end{cases} \quad (1)$$

Zero crossing rates (ZCR) are often calculated in the ASR community over a period of 10 ms [15]. Two uses of ZCR are to distinguish speech from background and to find starting points or endpoints of isolated sounds [16, 17]. Zero crossing rates are correlated to the spectral centroid feature and dominant frequencies features [14, 18]. For

the vibration data, zero crossing rates were calculated for approximately every 200 ms, or once per window. As previously stated, each window is 2,048 points. Figure 7 illustrates a 10 ms segment of data with the corresponding zero crossing rates.

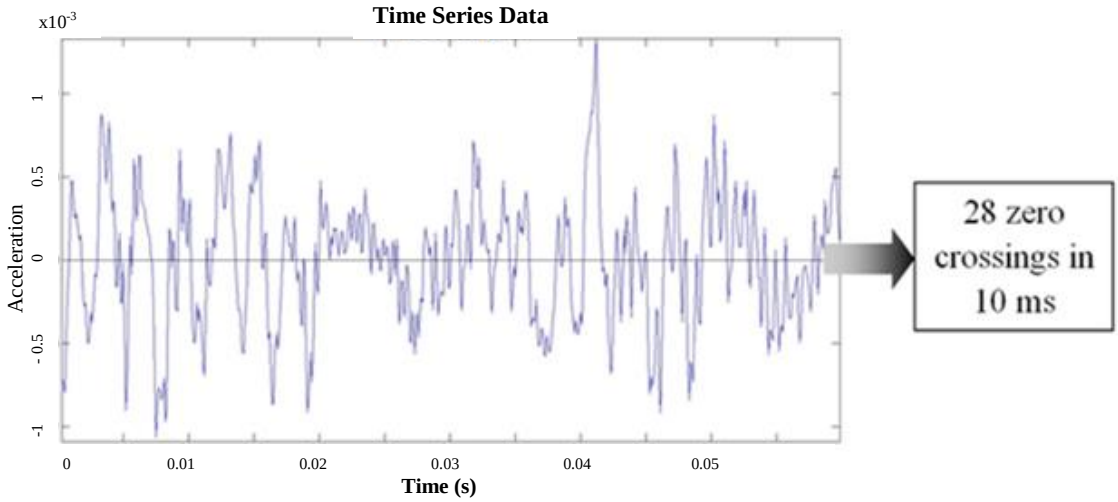


Figure 7. Time-domain data sample and the resulting ZCR.

Complexity: Complexity is a seismology feature that compares seismic energy from different parts of seismic waves. One use of complexity is to distinguish between earthquakes and explosions. The energy in the first 5 seconds is divided by the energy in the following 30 seconds of a seismic wave. Earthquakes often have higher complexity values than explosions as they have longer complex waveforms [19, 20]

$$C = \sum_{0}^{307} x^2(n) \,/\, \sum_{308}^{2048} x^2(n) \tag{2}$$

Complexity is calculated using Equation 2 where x is the time series data [19]. Since primary waves are not found in vehicles, a ratio of the energy in the first 15% of the data is divided by energy in the rest of the signal to find complexity for the vibration data. Complexity is calculated once per window by finding the energy in the first 307 points and dividing by the energy in the remaining 1,741 points. Figure 8 shows a scatter plot of complexity values for idle and constant rpm data for six different classes.

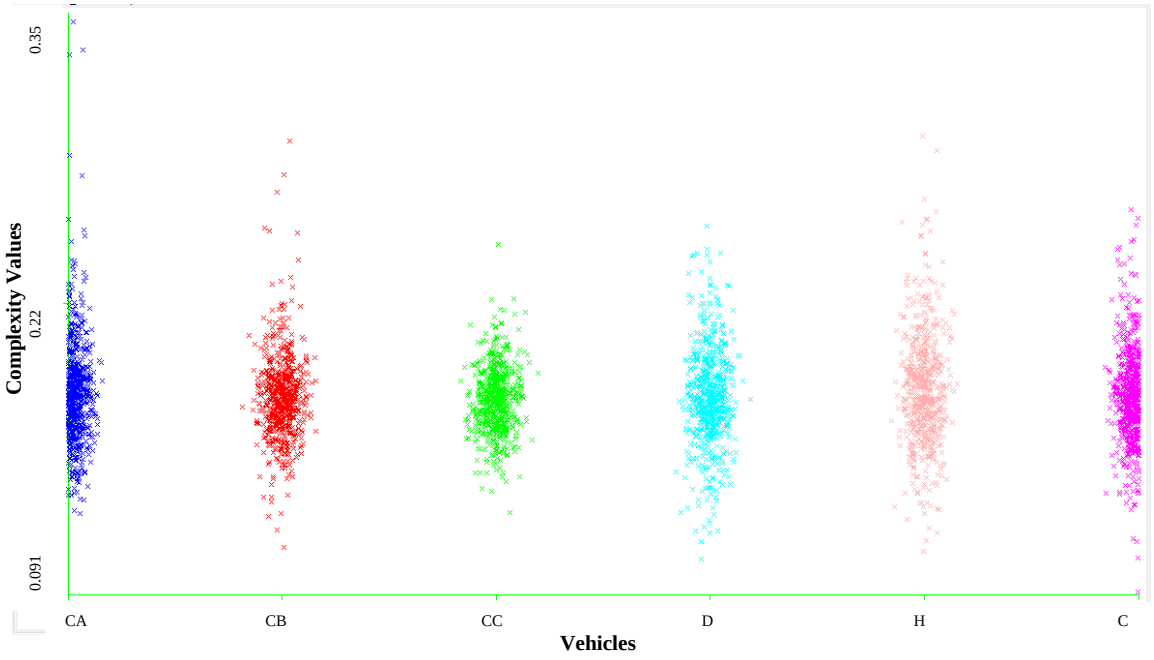


Figure 8. Scatter plot of complexity for six classes from idle and constant rpm accelerometer data.

Root Mean Square (RMS): The RMS is an average intensity measure in a given signal. RMS is a feature that is used in the speech processing community. RMS has been used to determine the portion of frames with lower intensity levels by comparing the RMS of a specific frame, generally 10 ms, to the average RMS of the overall signal. This feature is calculated using Equation 3, where x is the signal and N is the total of number of points. For the vibration data, RMS is calculated for a window of data, approximately every 200 ms, and averaged every five windows. Raw data, not normalized data, is used to calculate RMS. Normalizing the data makes the RMS value the same for every calculation regardless of the data.

$$RMS = \sqrt{\frac{1}{N}\sum_{i=0}^{N-1} x_i^2} \tag{3}$$

Linear Prediction Coefficients (LPC): LPCs are used in the speech processing, biological signal processing, and seismic signal processing communities. An estimation of vocal tracts filter is represented by LPC [21]. The coefficients are generally used to model unknown information such as vocal tract or distribution of parameters for brain waves [22]. Covariance or autocorrelation methods are utilized to calculate linear prediction coefficients. Details on linear prediction are explained fully in [22].

LPCs are often converted to reflection coefficients or cepstral coefficients so they represent information as individual coefficients rather than a group of coefficients that are dependent upon each other. Equation 4 shows the standard linear prediction coefficient equation, where *a* is the prediction coefficient, *y(n-k)* is a past sample, and *p* is the number of linear prediction coefficients that must be less than the total number of samples. The LPCs are calculated using built-in MATLAB functions and converted to linear prediction reflection coefficients for training and testing.

$$y(n) = -\sum_{k=1}^{p} a_k * y(n-k) \tag{4}$$

Frequency-domain features

Dominant Frequency: Dominant frequency is a versatile feature used to find the frequency with largest magnitude or peak. Dominant frequencies are the frequencies of the *N* largest magnitude peaks. Three dominant frequencies are calculated to represent each vibrometry signature. Three dominant frequencies were calculated using Equation 5, where each point in the periodogram is compared to determine the frequencies with the largest magnitudes. Subband spectral centroids and inverse zero crossing intervals are also used to calculate dominant frequencies in the speech community [12]. Figure 9 shows an example of the domain frequencies for some vibration data.

$$DF = i, j, k \; where \begin{cases} \frac{|FFT_i|^2}{N} > \frac{|FFT_j|^2}{N} > \frac{|FFT_k|^2}{N} \\ \forall_{l \epsilon \frac{|FFT|^2}{N}} \left(\frac{|FFT_i|^2}{N} > \frac{|FFT_l|^2}{N}\right) \\ \forall_{l \epsilon \left\{\frac{|FFT|^2}{N} - \frac{|FFT_i|^2}{N}\right\}} \left(\frac{|FFT_j|^2}{N} > \frac{|FFT_l|^2}{N}\right) \\ \forall_{l \epsilon \left\{\frac{|FFT|^2}{N} - \frac{|FFT_i|^2}{N} - \frac{|FFT_j|^2}{N}\right\}} \left(\frac{|FFT_i|^2}{N} > \frac{|FFT_l|^2}{N}\right) \end{cases} \quad (5)$$

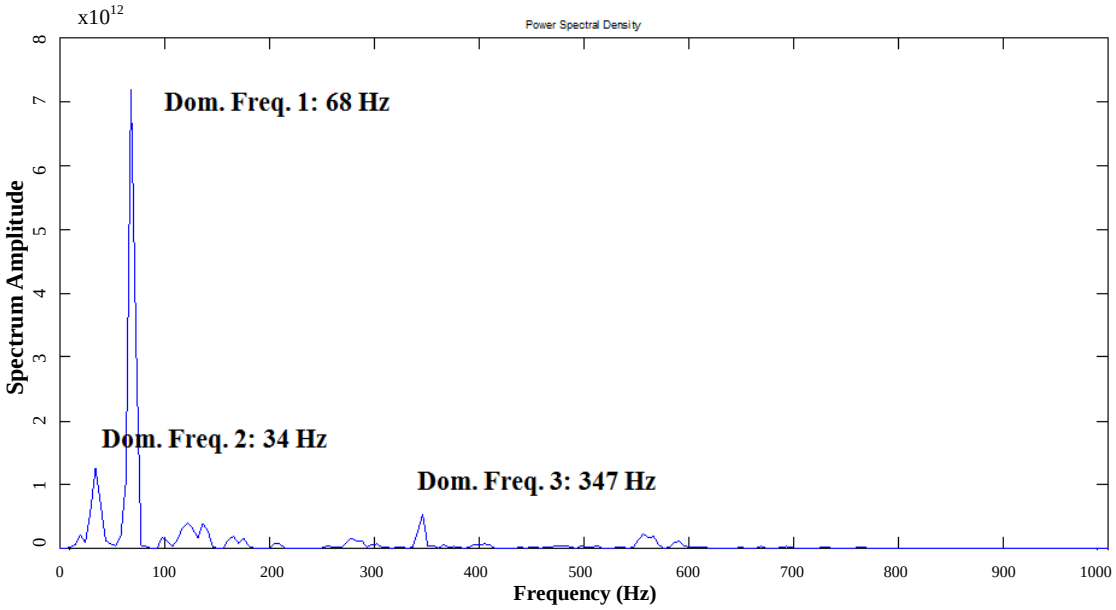


Figure 9. PSD of vibration data sample and corresponding dominant frequencies.

Flux (Spectral Flux): Spectral flux calculates the rate of change between frames of a signal. Flux is commonly used in speech processing. Spectral flux is calculated as a difference in magnitude of short time frequency transform (STFT) spectrums between frames that are normalized in energy [21]. This feature is used in [15] to distinguish between fast changing speech and steady changing music. Spectral flux is calculated using Equation 6, where *FFT* is a signal's spectrum, σ is standard deviation of the windowed spectrum, t is time, and N is the number of frequencies. Since there are 5 windows per second a data, a total of 4 flux values are created and averaged together for each vibration data trial.

$$F = \sum_{i=0}^{N-1} \left(\frac{\frac{|FFT_i(t)|^2}{N}}{\sigma(t)} - \frac{\frac{|FFT_i(t-1)|^2}{N}}{\sigma(t-1)} \right)^2 \quad (6)$$

Mel-Frequency Cepstral Coefficients (MFCC): MFCC are used in the speech processing community to convert speech into text. These coefficients represent a condensed spectrum motivated by the human auditory system. The compact representation of the auditory signal is why MFCC are so frequency used. Frequencies cannot easily be distinguished between each other by the auditory system when they are higher than 1000 Hz [21, 23]. MFCC are calculated by creating signal frames, windowing those frames (often, including here, using Hamming windows), computing the Fast Fourier Transform (FFT), applying a Mel-frequency filter bank, and then performing a Discrete Cosine Transform (DCT) [23].

Ten MFCC are calculated and used as features in this experiment following Equation 7. In the equation, N is the number of frequencies in the spectrum and FFT_{avg} is the signal's averaged spectrum. Approximately 200 ms frames were used, Hamming windowed, and each MFCC (0-9) is calculated using a filter bank with 3 overlapping triangular windows that each span all of the frame except for a few points. A scatter plot of MFCC0 values for six different classes is shown in Figure 10, where the different colors represent the different classes. This plot is a direct output from WEKA.

$$c(n) = \frac{1}{N} \sum_{k=0}^{N-1} \log |FFT_{avg}(k)| e^{j2\pi k \frac{n}{N}} \quad (7)$$

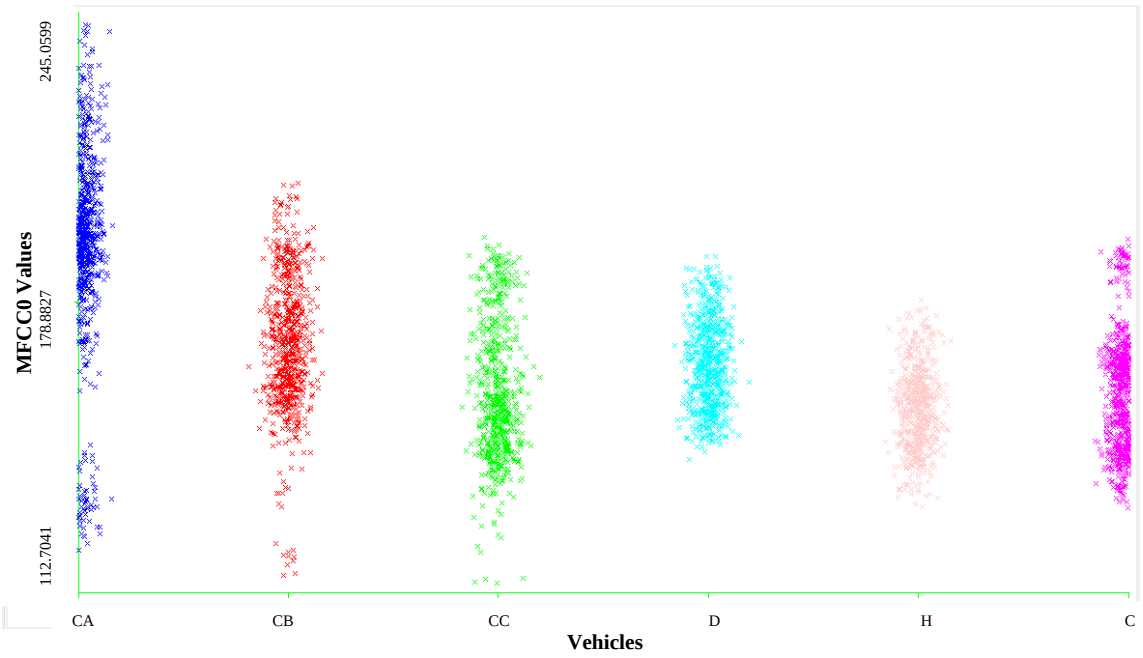


Figure 10. Scatter plot of MFCC0 for six classes from idle and constant rpm accelerometer data.

Peak Counter: Peak counting, or number of peaks, is a feature describing the number of frequencies with amplitudes higher than a selected threshold. Different algorithms and thresholding methods can determine which peaks are parts of the signal. One use of peak counting in speech processing is endpoint detection and segmentation of speech [24].

Equation 8 represents the method used for peak counting, where $\frac{|FFT|^2}{N}$ the periodogram of the signal, M is the number of frequencies, and T is an amplitude threshold. Peaks higher than the mean amplitude are counted for this experiment. Improved peak detection is a future goal for a more accurate peak count. Figure 11 shows a signal's PSD and the resulting peak count above the average peak amplitude.

$$NP = \sum_{i=0}^{M-1} \begin{cases} 1 & \frac{|FFT_i|^2}{N} > T \\ 0 & otherwise \end{cases}, T = mean(\frac{|FFT|^2}{N}) \quad (8)$$

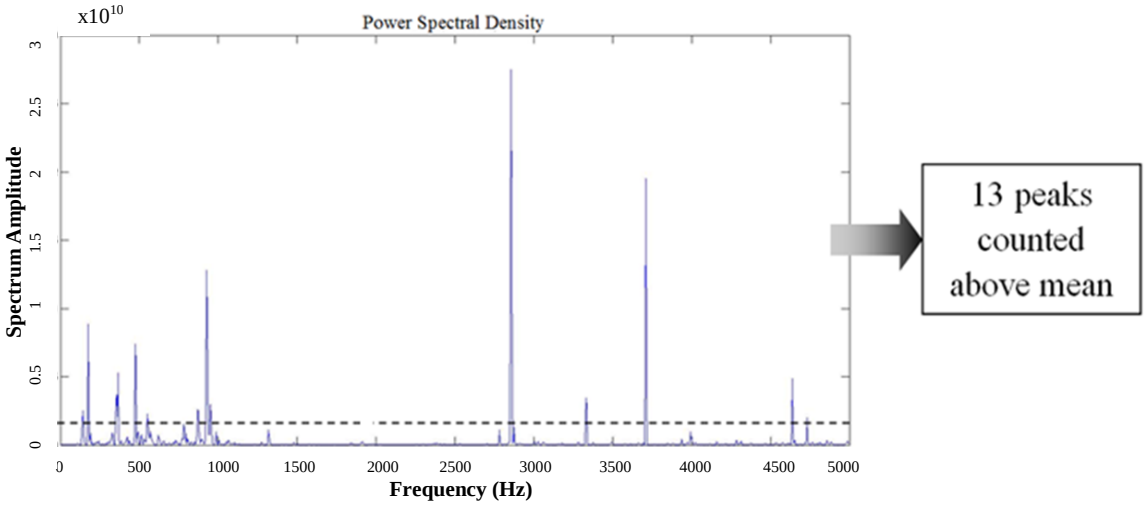


Figure 11. PSD of an idle and constant data sample from Figure 7 and the resulting peak count.

Rolloff (Spectral Rolloff): Spectral rolloff determines the frequency value where the spectrum reaches a certain percentage of the total energy of the spectrum. Rolloff is a feature that has a lot of possible applications. One application of spectral rolloff is to measure the skewness of a spectrum [15]. Rolloff is used to represent timbre characteristics in the music and speech processing communities [25].

Equation 9 is used to calculate spectral rolloff where $\frac{|FFT_i|^2}{N}$ is periodogram of the signal, r is the unknown spectral rolloff, N is the number of frequencies, and T is the percentage threshold. For this experiment the percentage threshold of rolloff is 90% of the spectrum's energy.

$$T = \sum_{i=0}^{r} \frac{|FFT_i|^2}{N} / \sum_{j=0}^{N-1} \frac{|FFT_j|^2}{N} \quad (9)$$

Spectral Centroid: Spectral centroid is another generic statistical feature. The spectral centroid of a signal represents the center point, or balancing point, in the spectrum. Spectral centroids are sometimes used in the speech community to describe the timbre in music or speech and can be used to distinguish voiced and unvoiced speech [15, 25]. By

creating bands within a spectrum the spectral centroid can also be used to find dominant frequencies.

This feature is calculated using Equation 10 where FFT is the spectrum of the signal and *N* is the number of frequencies [1]. Spectral centroid is calculated for the vibration data for every window (200 ms windows) and averaged over five windows. Figure 12 illustrates the lack of average difference between the vibrations of six different vehicles.

$$Sc = \sum_{i=0}^{N-1} i * \frac{|FFT_i|^2}{N} / \sum_{j=0}^{N-1} \frac{|FFT_j|^2}{N} \quad (10)$$

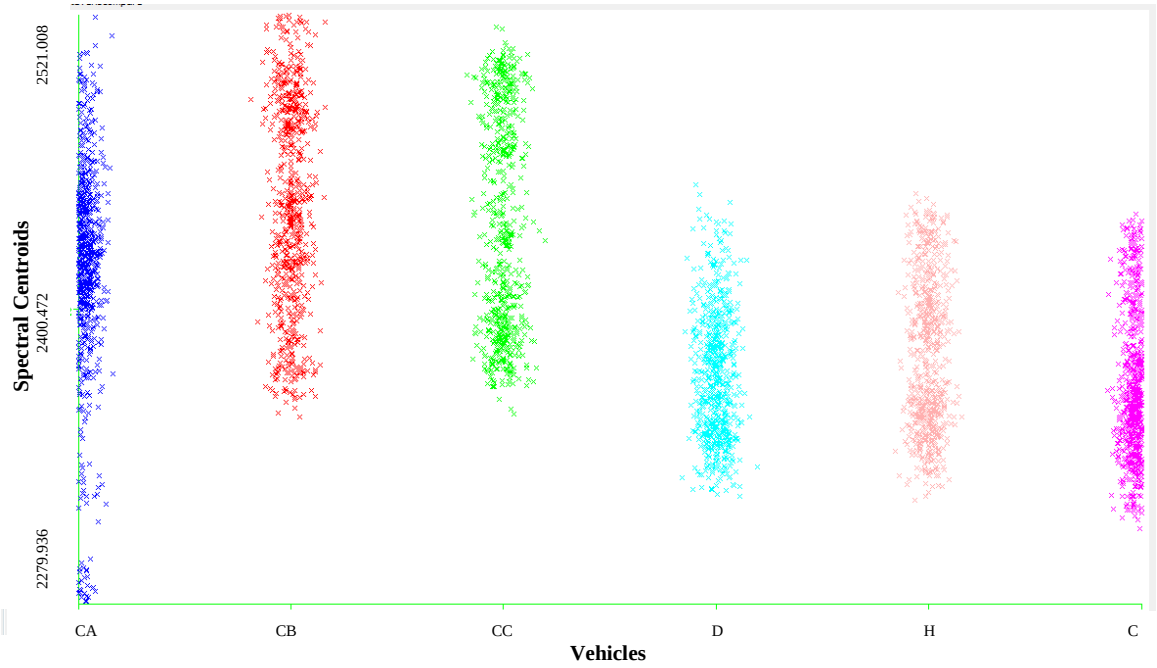


Figure 12. Scatter plot of spectral centroids for 6 classes of idle and constant rpm data.

Spectral Ratio: Spectral ratios are often used as a seismology feature, but have other possible applications as a spectral feature. Spectral ratios compare frequency amplitudes between two different bands such as low vs. high frequency. In [19], specific ratios are used to decide if a seismic wave is from an earthquake or it is due to an explosion.

For this experiment, three arbitrarily selected spectral ratios were used to determine the ratio of frequency magnitudes using Equation 11, where X is a signal's spectrum and

x_0, x_1, and x_2 are the frequencies that divide the bands, where $x_0 < x_1 < x_2$. The three ratios were created for each window using: (1) 0-100 Hz over 100-5000 Hz, (2) 0-500 Hz over 500-5000 Hz, and (3) 0-1000 Hz over 1000-5000Hz.

$$Sr = \sum_{i=x_0}^{x_1} \frac{|FFT_i|^2}{N} / \sum_{i=x_{1+}}^{x_2} \frac{|FFT_i|^2}{N} \quad (11)$$

3.3. Feature Selection

Feature selection techniques are often implemented to remove features that do not help target separation or result in redundant information. Feature selection reduces data dimensionality in a way that does not alter the original data [26]. Sequential forward selection (SFS) and ReliefF, described in [26] and [27], respectively, are two feature selection filters that were used to rank and select features from the 33 features previously extracted. Filters determine a feature's score based on fundamental properties of the data, such as first order statistics [26]. The score is used to remove low ranking features. SFS was used because it provides a user friendly baseline technique. ReliefF was utilized previously as a feature selection method for vibrometry data [1]. ReliefF is a reliable and complete selection technique for incomplete and/or noisy data.

Sequential forward selection begins sorting features by creating an empty set to add features that create the highest classification rates. The best feature is selected using some criteria value and added to the empty set to maximize classification using a single feature. Figure 13 illustrates the pseudo code for the sequential forward selection technique, where k is the number of features, Y is the set of best features, x^+ is the next best feature, and $J(Y_k + x^+)$ is the criteria value [26]. Euclidean distance, shown in Equation 12, was used as the criteria value in this work [28]. The two variables $\boldsymbol{p}$ and $\boldsymbol{q}$ represent the

vectors of the *k* best features being tested for two vibration signatures. The number of features used in the distance equation, *k*, will increase as the best features are selected.

$$d(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{\sum_i (p_i - q_i)^2} \tag{12}$$

Sequential Forward Selection (SFS) Pseudo Code

1. Create an empty set: $Y_k = \{\emptyset\},\ k = 0$.
2. Select best remaining feature: $x^+ = arg\ max_{x^+ \in Y_k}[J(Y_k + x^+)]$
3. If $J((Y_k + x^+) > J((Y_k)$
    a. Update $Y_{k+1} = Y_k + x^+$
    b. $k = k + 1$
    c. Go back to step 2.

Figure 13. Pseudo code for sequential forward selection from **[13]**.

Using Euclidean distance, the feature selector works similarly to a K-nearest neighbor classifier, which is explained in more detail later [28]. The distances between a point and its k, which when using WEKA is 49, closest points are calculated for each feature. The feature that creates the greatest average distance between different classes is selected as the best. Features are continually added to the set to improve the overall separation of classes until the separation can no longer be improved. When the performance is maximized, only the features selected as the best are kept to be used for classification. SFS performs the best when the number of features that separate classes is small. One downside is that once features are added to the feature set they cannot be removed using the baseline SFS [26]. The suboptimal results often created by this technique led to adjustments and pruning steps to improve this algorithm. These improved SFS techniques are not tested in this work because the goal was to find the baseline performance as a starting point in this end-to-end process.

ReliefF is a feature selection method derived from the original Relief algorithm presented in [29]. The original Relief feature selection approach found statistically useful Boolean and numerical features in a two-class problem. The original algorithm was sensitive to noisy and incomplete datasets so improved algorithms like ReliefF were created to improve versatility. ReliefF gives each feature a ranking value to determine the highest ranking features [27]. The pseudo code for ReliefF is illustrated in Figure 14.

ReliefF Pseudo Code

1. Set weights for all features (A): W[A]=0
2. For all instances
    a. Select a random instance, R_i
    b. Find k nearest hits, H_j
    c. For all other classes ($class\ C\ \neq class\ R_i$)
        i. Find k nearest misses for each class, $M_j(C)$
    d. For all features
        i. Update weights: $W[A] = W[A] - \frac{\sum_{j=1}^{k} diff(A,R_i,H_j)}{mk} + \frac{\sum_{C \neq class(R_i)} \frac{P[C]}{1-P[class(R_i)]} \sum_{j=1}^{k} diff(A,R_i,M_j(C))}{mk}$

Figure 14. Pseudo code for ReliefF feature selector **[13]**.

Each weight value, *W[A]*, is set to zero to start with. A random sample, called *R*, is automatically selected. *K* nearest neighbors, found using Mahalanobis distance, in the same class called 'nearest hits', or *H*, are found for each feature, *A*. The process is repeated to find *k* nearest neighbors in each of the other classes. The neighbors in these classes are called 'nearest misses', or *M(C)*, because the values are close to the instance but not in the same class. The weight is calculated and updated with each sample and each sample's nearest hits and nearest misses using Equation 13 and 14 [27]. The features are ranked from best to worst based on this weight. Generally a threshold is set

to pick features until a certain classification rate is achieved or a percentage of features are selected. For this work, the threshold of 5 features was selected [1].

$$W[A] = W[A] - \frac{\sum_{j=1}^{k} diff(A,R_i,H_j)}{mk} + \frac{\sum_{C \neq class(R_i)} \frac{P[C]}{1-P[class(R_i)]} \sum_{j=1}^{k} diff(A,R_i,M_j(C))}{mk} \quad (13)$$

$$diff(A,R_i,H_j) = \frac{|R_i - H_j|}{\max(A) - \min(A)} \;\&\; diff(A,R_i,M_j(C)) = \frac{|R_i - M_j(C)|}{\max(A) - \min(A)} \quad (14)$$

3.4. Classification

Classifiers are used to determine the separability of targets. Using features, classifiers were trained and tested to understand the performance of the end-to-end process. Confusion matrices, kappa statistics, and receiver operator characteristic (ROC) curves were selected as the primary performance metrics to simply illustrate and understand the classification results. Confusion matrices provide a means to quantify the classification algorithms performances. Kappa statistics are a measure of classification confidence showing the magnitude of agreement between classification decisions represented as a score between 0 and 1 [30]. Values closer to one suggest higher confidence in the classification results; whereas, values closer to zero suggest poor confidence in the classification results. An acceptable kappa statistic is generally considered to be anything over 0.8. ROC curves illustrate classifier performance as a comparison of true positive classification rates vs. false positive classification rates. The closer the area under the ROC curve integrates to one the more successful the classifier is considered to be.

The four classifiers used were based on previous classifier selections in vibrometry analysis. Naive Bayes, decision tree, support vector machines (SVM), and k-nearest neighbors (KNN) were selected to start with based on the work in [1] and [5]. Each

classifier decides the target's class in mathematically different ways offering diversity amongst the classifiers. Naive Bayes, KNN, and decision trees are all well-established baseline techniques. SVM is a newer method of classification that was only used once with vibrometry data previously [5]. Training and testing the data was completed using the 10-fold cross validation method. The technique separates the data into ten equal subsets. Nine of the subsets are used for training the classifier and the 10^{th} is used for testing. The process is repeated until every subset is selected as the testing set. All of the results are averaged together for final result [5]. Classification using 10-fold cross validation was completed because the datasets that were analyzed were both small. This classification separation was not the only data separation technique used. For a few of the trials classified, a training and a testing set where used instead of 10-fold cross validation. For example, one run of the classifiers was done by training the classifiers with idle data and then testing the classifier with constant revolution per minute (rpm) data.

Naive Bayes

The Naive Bayes classifier is a probabilistic classifier. Other Bayesian classifiers have large computational complexities. Naïve Bayes makes strong conditional independence assumptions that are used to simplify the classification process [31]. Equation 15, which is Bayes Rule, is used to calculate the probability that a specific vibration signature in class Y belongs to some class y_k given the features, X_i. Some parameter estimations are required to use the Naïve Bayes classifier because the true class of the target's vibration signature is not known. The prior probability of Y and the probability that the feature X_i takes on a specific discrete value x_{ij} given that $Y=y_k$ are two values that require estimation. These two probabilities, θ_{ijk} and π_k, are estimated using

Maximum Likelihood Estimation (MLE). With the estimations, Equation 15 can be solved and the class with the highest probability is most likely to be the true class of the target.

$$P(Y = y_k | X_1 \dots X_n) = \frac{P(Y=y_k)P(X_1 \dots X_n | Y=y_k)}{\sum_j P(Y=y_j)P(X_1 \dots X_n | Y=y_j)} = \frac{P(Y=y_k)\prod_i P(X_i | Y=y_k)}{\sum_j P(Y=y_j)\prod_i P(X_i | Y=y_j)} \quad (15)$$

Training and testing occur very quickly because Naïve Bayes is simplified with the independence assumptions. Its training time complexity is *O(MN)*, where *M* is the number of samples and *N* is the number of features [31]. Naïve Bayes classifies more successfully when independence between features does exist. Redundancies and overlapping features degrade the performance of Naïve Bayes. As previously stated, the classification is done within WEKA. Naïve Bayes is just referred to as [NaiveBayes] in WEKA.

K-Nearest Neighbor (KNN)

K-nearest neighbor is non-parametric classification algorithm. KNN searches for each feature vector's *k* nearest neighbors using a distance measure, usually the Euclidean distance. Equation 16 is used to calculate Euclidean distance, where p_i and q_i denote the i^{th} feature value for two vibration feature vectors, *n* is the number of features, and *d* is the distance measure [32]. Using all of the feature values of a sample, the distance between two vibration feature vectors is calculated; all 33 feature values or the features selected in the previous section of the end-to-end process are used to calculate the distance. A positive *k* value of 3 was chosen for this research to produce similar results to [1] and [5]. The distance between each sample's feature set and the target's feature set are calculated. The *k* smallest distances are selected as the nearest neighbors. The target's class is found by majority vote using the *k* nearest neighbors' classes.

$$d(\boldsymbol{p}, \boldsymbol{q}) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots (p_n + q_n)^2} \quad (16)$$

KNN is very slow to test data because the only task accomplished during the training period is to save all of the data. The target data has to be compared to every point to determine its *k* nearest neighbors. The testing time complexity for KNN is $kM(O(kM))^2$ where *k* is the number of neighbors compared to and *M* is the number of training samples. Inside of WEKA, the KNN algorithm used for this work is called [IBk]. WEKA uses instance based learning, or modified KNN, to allow for the voting to occur later if more information is required to increase correct classification rates.

Decision Trees

Making decisions by branching into a tree-like model is known as decision tree classification. The decision tree is formed by determining which features maximize the classification decision. Using a gain equation, the significance of branching is found to determine how well branches separate the data. If the significance of the branch is greater than a given threshold, the branch is added. Different branching thresholds and information removal methods exist to provide the most accurate and timely classification.

For this work, the C4.5 method was chosen to implement based on the suggestions from WEKA used in [5]. C4.5 classifies targets using divide-and-conquer theory [33]. The distribution of the dataset, *p(D, j)*, is calculated to determine the number of vibration signatures, *D*, that belong to each class, *j* [34]. If all the data belongs to the same class, there is not enough data to proceed. Otherwise, the information and gain are calculated using Equations 17 and 18. The information sums up the products of the distributions of the data and the log of the distributions over the total number of classes, *C*. The gain is calculated for a specific test, *T*, by subtracting the sum of all the information of the tests

from the information from the overall data. The tests examine how a feature can be compared to a threshold to separate classes. For example, comparing spectral centroid to a threshold and examining the separation of classes would be a test. The test has T_i mutually exclusive results that are used to partition the data into k subsets called D_i [33]. The decision to split the tree or stop constructing the tree is determined by the quotient of the gain and split equations, shown in Equation 18 and 19.

$$Info(D) = -\sum_{j=1}^{C} p(D,j)\log_2\big(p(D,j)\big) \tag{17}$$

$$Gain(D,T) = Info(D) - \sum_{i=1}^{k} \frac{|D_i|}{|D|} * Info(D_i) \tag{18}$$

$$Split(D,T) = -\sum_{i=1}^{k} \frac{|D_i|}{|D|} * log_2\left(\frac{|D_i|}{|D|}\right) \tag{19}$$

All of the possible splits are tested. If the gain ratio, the quotient of the gain and split equations, increases above a threshold the split is considered. The test with the highest gain ratio is selected for the split [33]. The tests are repeated until the gain ratio reaches a threshold which is usually close to zero. The construction of the tree stops when this threshold is met. For WEKA, if the gain ratio is lower than 0.25 the construction of the C4.5 decision tree, called [J48], ends. The decision tree is created only using the training data. The training time complexity of C4.5 is $O(MN^2)$, where M is the number of training samples and N is the number of features or dimensions.

Support Vector Machines

Support vector machines (SVM) are a more recent linear classifier, with many non-linear applications and algorithms. This classifier works by mapping features to a higher dimensionality, uses a linear or non-linear kernel function to create hyperplanes, and maximizes distance between the data and the hyperplane [35]. Polynomial kernels, radial basis kernels, and sigmoid kernels are just a few of the kernel functions SVM can use

[36]. One problem with SVM is that it needs to solve large quadratic programming problems.

Non-linear SVM is computed from the Lagrange multipliers in Equation 20. The output of the SVM, *u*, is found by summing the products of y_j, +1 if the feature vector is on the correct side of the hyperplane and -1 otherwise; the Lagrange multipliers α_j; and the kernel function $K(x_j, x)$ where x_j is a training vibration signature and x is an inputted vibration signature [37, 38]. The threshold b is subtracted from the sum. In WEKA, the kernel used for the SVM, called [SMO], is the polynomial kernel with exponent $d=1$.

$$u = \sum_{j=1}^{N} y_j \alpha_j K(\overrightarrow{x_j}, \vec{x}) - b, \; K\left(\overrightarrow{x_j}, \vec{x}\right) = < \overrightarrow{x_j}, \vec{x} >^{d} \tag{20}$$

The quadratic problem that needs to be optimized is shown in Equation 21, where α is the vector of N non-negative Lagrange multipliers, L_D is the dual Lagrangian function to be maximized, x is a training vibration signature, y is +1 or -1 depending on if x is corrected divided by the hyperplane or not, and C is a tuning parameter [37]. C is set to 1 in WEKA. The problem is considered solved when the Karush-Kuhn-Tucker (KKT) conditions, shown in Equation 22, are fulfilled. KKT conditions are only satisfied when the variables α_i are optimized. Three methods to optimize the variables are briefly covered in [38] with reference to the original papers for more details. The idea of the optimization is to keep track of the KKT conditions that need to be met, to find a strategy to encroach on the optimum by increasing L_D while respecting its constraints, and finding a way to analyze only a portion of the data at a time [38].

$$\begin{gathered} \max \boldsymbol{\alpha} \left(L_D = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j K\left(\overrightarrow{x_i}, \overrightarrow{x_j}\right) \alpha_i \alpha_j \right) \\ Subject\ to: 0 \le \alpha_i \le C, \forall i \\ \sum_{i=1}^{N} y_i \alpha_i = 0 \end{gathered} \tag{21}$$

$$\begin{gathered} \alpha_i = 0 \rightarrow y_i f(\overrightarrow{x_i}) \geq 1, \\ 0 < \alpha_i < C \rightarrow y_i f(\overrightarrow{x_i}) = 1, \\ \alpha_i = C \rightarrow y_i f(\overrightarrow{x_i}) \leq 1 \end{gathered} \tag{22}$$

Solving these problems takes a lot of time and makes the classification process slow. In [37], a sequential minimal optimization (SMO) was proposed as a method to simplify solving SVM. Rather than solve a complex optimization problem, small optimization problems that can be solved analytically are utilized for SMO [37]. Solving analytically speeds up the classification process dramatically because iterating is unnecessary. Training the SMO algorithm requires a lot of time but testing data occurs relatively fast. The training time complexity of SVM is $O(M^3)$ while SMO's time complexity ends up between $O(M) - O(M^{2.3})$ where M is the number of training samples.

3.5. Analysis

As previously stated, confusion matrices and kappa statistics are created to analyze the classification results. The results from each of the confusion matrices and kappa statistics are analyzed as a human in the loop decision process. Based on the performances, changes are decided by the human in the loop to improve the overall end-to-end process. Decisions are made on how the different sets of features perform, which features are continually selected by the feature selection filters, and how classifiers perform. If only a specific feature is continually selected then that feature would be kept and the rest of the features replaced with new ones. These decisions continue to be made until the best possible outcome, consistent and accurate classification results, is achieved by the end-to-end process.

4. DATA

4.1. Mountain-top/Airborne Long-Range Test & Evaluation (MALTESE)

MALTESE is a dataset that was collected by the Sensors Directorate of the Air Force Research Laboratory (AFRL). The dataset was collected on 3 separate military vehicles on desert terrain using contact accelerometers and a long range laser vibrometer, but this research utilizes only the contact sensors. The three military vehicles are referred to as AA, BA, and CA. The data was collected before many laser vibrometer data collections had occurred. The data on the three vehicles with a mix of piston and turbine engines was collected while stationary and moving; however, only stationary data was used. The following stationary vehicle engine states were recorded: idle; engine revving, or sweeping, or changing continuously, through the engine's revolutions per minute (rpm); and different constant rpm levels.

MALTESE was collected over several days in the spring using single axis and tri-axial contact accelerometers spread across the vehicles' surfaces. The data was collected with a sampling rate of 10 kHz. A few dozen trials were performed each collecting 30 second durations of data. However, not all of the data collections were fit for analysis. Only 'good quality' stationary data from the collection could be used for analysis. Good quality refers to data that has been corrected for errors including noise spikes, saturation, and missing data points. Each vehicle provided 4140-one second samples of quality

stationary data. A total of 12420 one second samples of idle and unspecified, constant rpm MALTESE data were used for analysis of the end-to-end process.

4.2. Laser Vibrometer & Accelerometer Simultaneous Collection (LAVA-SC)

LAVA-SC is a multi-sensor dataset collected as a part of this thesis effort. Contact tri-axial accelerometers The same three acquisition boxes used in MALTESE were used to collect the accelerometer data. Each acquisition system supported ten tri-axial accelerometers. A total of 30 accelerometers were used to analyze each vehicle. The laser vibrometer used to collect data was a Polytec OFV 503 sensor head and Polytec OFV-5000 acquisition box. The acquisition systems and representative sensors used are shown in Figure 15.

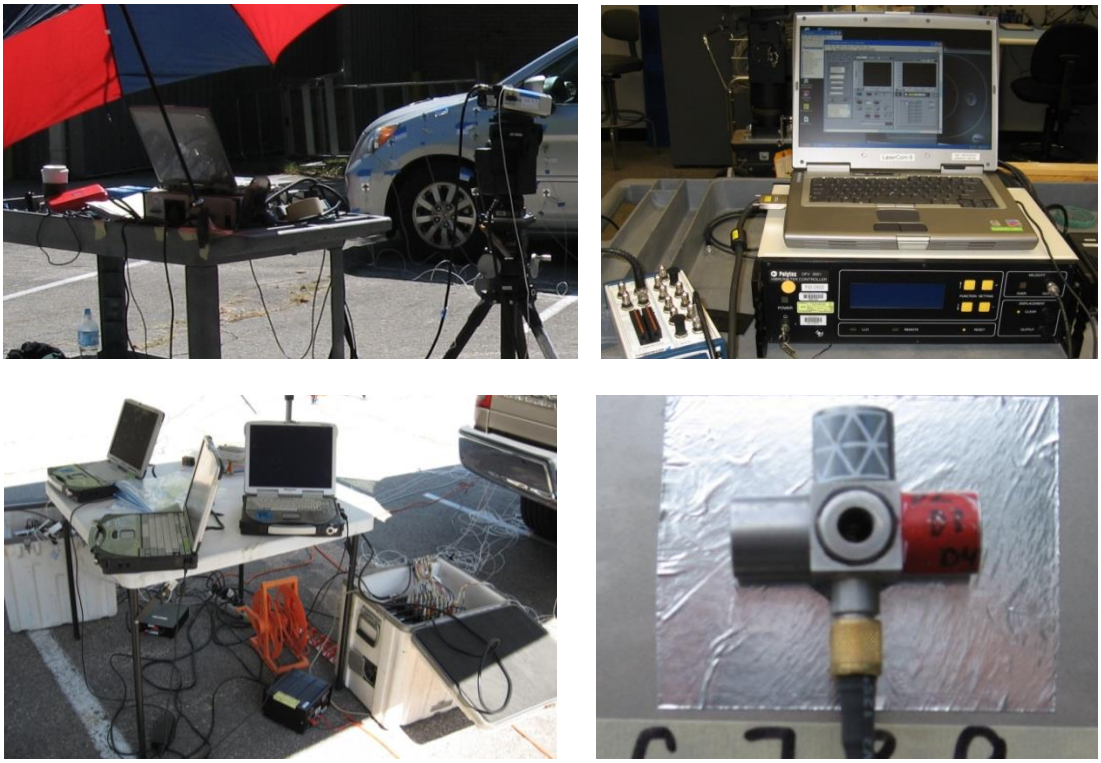
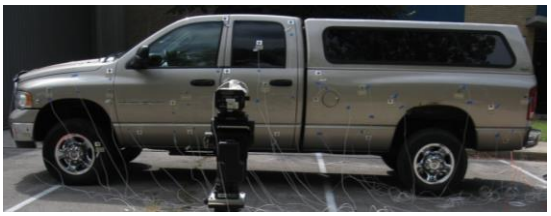


Figure 15. Images of the sensors and acquisition systems from the LAVA-SC data collection.

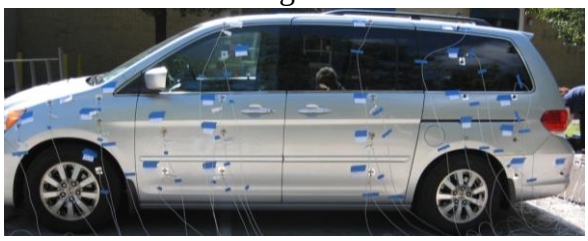
A total of five collections took place over a week during the autumn in a parking lot away from traffic and other noise sources. The four vehicles that were collected include: (1) 2004 Chevrolet Cavalier; (2) 2003 Dodge Ram 2500, which was collected on two separate days; and (3)/ (4) 2010 Honda Odyssey EX-L which were the same make and model but different serial numbers. These four vehicles are shown in Figure 16. Three classes were used from the LAVA-SC dataset: D (Dodge), H (Honda), and C (Chevy). The Dodge Ram data from the two collections were combined to expand the number of operating conditions the vibration data was exposed to. The Honda Odyssey data was also combined for use during these experiments as the two vehicles should be nearly identical.

2004 Chevrolet Cavalier



2003 Dodge Ram 2500

2010 Honda Odyssey EX-L



2010 Honda Odyssey EX-L

Figure 16. Images of the vehicles from LAVA-SC collection.

Stationary vehicle data included idle, constant engine rpm settings, rpm sweeping, and turning the vehicle off was collected on the vehicles. However, only idle and constant rpm data at 50% of the vehicle's maximum rpm, 1500 rpm for the Dodge Ram and 3500 rpm for the other vehicles, was used. Accelerometers were mostly placed on the driver

side of the vehicles with a few reference locations on the other sides of the vehicles. Concentrating accelerometers on one side of the vehicle was done to collect more points on the vehicle simultaneously using the laser vibrometer. The laser vibrometer collected pinpoint spots on the accelerometers. Several dozen accelerometer and laser vibrometer trials of data were collected for 15 second increments. Each vehicle had over 6,000 one second samples of quality idle and constant rpm accelerometer data and 500 one second samples of laser vibrometer data. A total of 19,000 one second samples of laser vibrometer and accelerometer LAVA-SC data were used to analyze the end-to-end process.

5. EXPERIMENTS

As previously stated, MATLAB and WEKA were used to implement the end-to-end process shown in Figure 3. MATLAB was used to preprocess data and extract features. WEKA was used to select features and classify the data. Each 15 and 30 second collection of data from MALTESE and LAVA-SC were broken into one second segments. A single collection of data became 15 LAVA-SC data samples and 30 MALTESE data samples. Conversion to velocity was performed for the accelerometer and laser vibrometer data from LAVA-SC. All of the samples were windowed to 2,048 points with 25% overlap, zero centered, and normalized prior to feature extraction. A PSD was also taken for all the windows of each sample for the frequency domain feature extraction. After the data was preprocessed, 5 feature sets, with one feature set per window, were calculated for each one second sample and averaged to create a single feature vector.

After the features were selected, three different runs of the end-to-end process were completed for each classifier. Running the end-to-end process all the way through with no feature selection, i.e. using all of the features, was completed first as a baseline. Completing the process with all of the features gave a general idea of the best average performance of the classifier using all available information. The process was completed again using each of the feature selection methods to determine if the overall classification rates changed in the confusion matrices. The performance improvements are attributed to

removal of redundant feature information and performance degradations to an under-utilization of feature information. Each classifier was used with all default parameter settings designated by WEKA. These parameters were used to ensure repeatability of the results and to avoid over tuning, or building a system to specifically fit a single set of data. The classification results were evaluated using confusion matrices and kappa statistics. When the sum of diagonals of the confusion matrix was greater than 85%, the classification was considered successful. A classifier's kappa statistic was considered acceptable when 0.8 or higher and excellent when it exceeded 0.95. Receiver operating characteristics (ROC) curves are generated using WEKA to provide a visual indication of each target class's classification performance.

5.1. Feature Extraction Statistics

Before classifying the data, some first order statistics including mean, standard deviation, minimum, and maximum, were calculated about each feature. These statistics were used as an initial investigation into which features provided target separability. The first order statistics of each feature was calculated to illustrate the distribution of feature values for each class. Table 1 shows several of the features and the calculated means, standard deviations, and ranges for MALTESE and LAVA-SC data.

Table 1. First order statistics for some features extracted from MALTESE and LAVA-SC results.

Feature	Mean LAVA-SC	Mean MALTESE	Std. Dev. LAVA-SC	Std. Dev. MALTESE	Min., Max. LAVA-SC	Min., Max. MALTESE
Zero Crossing	143.03	551.98	67.27	257.54	17, 342	34, 1430
Complexity	0.18	0.18	0.02	0.02	0.09, 0.37	0.08, 0.36
RMS	0.16	0.01	0.02	0	0.06, 0.29	0.003, 0.011
Dom. Freq. 1	125.77	785.82	118.43	832.18	10.73, 823.74	4.88, 4230.96
Flux	178.10	434.98	174.12	302.50	0.02, 1521.41	0.07, 1565.12
LPC0	-0.98	-0.87	0.02	0.11	-1, -0.89	-1, -0.38
LPC1	0.10	-0.15	0.11	0.23	-0.19, 0.57	-0.93, 0.61
LPC2	-0.02	-0.17	0.07	-0.19	-0.51, 0.18	-0.86, 0.46
MFCC0	163.76	173.45	11.24	17.77	132.79, 195.19	112.7, 251.45
MFCC1	1.78	-3.30	3.85	3.89	-11.81, 12.07	-13.14, 11.49
MFCC2	-3.26	-2.80	2.69	5.72	-12.48, 4.41	-12.23,5.83
Peak Count	52.21	63.01	9.34	11.13	30, 87	32, 97
Rolloff	461.52	2375.47	284.24	1091.46	24.4, 1367.38	63.44, 4869.26
Spec. Centroid	2369.51	2432.53	28.05	43.04	2309.22, 2452.69	2277.98, 2525.89
Spec. Ratio 1	3.55	0.61	8.57	2.39	0.003, 143.33	0, 39.95

Using these first order statistics, an indication of what features may be selected by the feature selection techniques was established. For example, zero crossing rates were very different between the military vehicles in MALTESE and civilian vehicles in LAVA-SC. The military vehicles have more zero crossings on average than the civilian vehicles. This separability indicates that this feature could be useful in separating military and civilian vehicles; however, the limited number of targets and specifically the lack of similar class vehicles in the two sets of data, prevent generalization of this result. In contrast, complexity would not be a very good feature for this data because the mean and standard deviation were identical. A few other examples of potentially useful features were dominant frequency, RMS, MFCC0, flux, and rolloff. The mean and standard deviation for complexity, LPC0, and spectral ratio 1 display a lack of separability

between the two datasets. Based on these statistics, these features would most likely not be selected by the feature selectors to separate the target classes. The outcomes of these features based on the first order statistics is explain in Section 5.5. It should again be noted that these features were not developed for this application and that one of the purposes of the development of this end-to-end process is to support future development and evaluation of optimized features.

5.2. MALTESE Results

Data from the MALTESE collection was sorted and analyzed in the end-to-end process three different ways: using idle accelerometer data, idle & constant rpm data, and piston engines vs. turbine engines. Idle accelerometer data was analyzed using the end-to-end process first. The end-to-end process was also tested on a combination of idle and constant rpm accelerometer data. The final set of data analyzed was piston vs. turbine data. The three classes were separated into their respective engine types to complete the piston vs. turbine analysis. These three trials were used to demonstrate the method to: (1) determine how robust the features were for different operating conditions; (2) determine how well the end-to-end process separated the three military vehicles; and (3) determine the separability of piston and turbine engines.

5.2.1. Idle Accelerometer Results

The first set of results was found by only using idle accelerometer data from the MALTESE collection. A total of 2,160 idle data samples, 720 samples per class, were ran through the end-to-end process without feature selection and using each of the feature selectors. Classifications with all features in KNN-3 and SVM produced perfect classification results. All of the classifiers had a kappa statistic greater than 0.9 when

using all the features extracted. When feature selection was used, SFS selected 7 features and the top 5 ReliefF features were used to classify the idle data. The features selected by SFS were: MFCC0, MFCC3, MFCC8, LPC2, LPC7, peak count, and dominant frequency 1. ReliefF selected MFCC3, peak count, MFCC0, MFCC8, and MFCC4. While using the ReliefF and SFS selected features the classification performance degraded slightly regardless of which classifier was used. All of the classification performances had kappa statistics higher than 0.85. The confusion matrices from running the end-to-end process using all of the features and feature selected by ReliefF and SFS are shown in Table 2, where ReliefF is in green and SFS is in red.

Table 2. Confusion matrices from using idle MALTESE accelerometer data.

Naïve Bayes

All Feature	AA	BA	CA
AA	98.75	1.25	0
BA	13.67	81.94	1.39
CA	0	0.42	99.58

Naïve Bayes

ReliefF	AA	BA	CA
AA	98.89	1.11	0
BA	8.89	90.83	0.28
CA	0	1.25	98.75

Naïve Bayes

SFS	AA	BA	CA
AA	96.67	3.33	0
BA	13.19	83.56	3.75
CA	0	11.25	88.75

KNN-3

All Feature	AA	BA	CA
AA	100	0	0
BA	0	100	0
CA	0	0	100

KNN-3

ReliefF	AA	BA	CA
AA	99.72	0.28	0
BA	0	100	0
CA	0	0	100

KNN-3

SFS	AA	BA	CA
AA	99.86	0.14	0
BA	0	100	0
CA	0	0	100

Decision Trees

All Feature	AA	BA	CA
AA	99.86	0.14	0
BA	0.14	99.58	0.28
CA	0.14	0.28	99.58

Decision Trees

ReliefF	AA	BA	CA
AA	99.31	0.69	0
BA	0.42	99.31	0.28
CA	0	0.28	99.72

Decision Trees

SFS	AA	BA	CA
AA	98.61	1.39	0
BA	0.69	99.03	0.28
CA	0.14	0.14	99.72

SVM

All Feature	AA	BA	CA
AA	100	0	0
BA	0	100	0
CA	0	0	100

SVM

ReliefF	AA	BA	CA
AA	99.03	0.97	0
BA	6.25	93.75	0
CA	0	0	100

SVM

SFS	AA	BA	CA
AA	91.25	8.75	0
BA	9.03	90.83	0.14
CA	0	0	100

This same set of data was used in [5] to test manifold learning techniques using these same four classifiers. Their process involved normalizing the data and classifying one second samples of data without features. The results seen in [5] without using feature extraction achieved classification rates around 50%, except for KNN-3. Using feature extraction has improved the classification rate significantly, as shown in Table 2. The average classification rate using feature extraction was over 89%.

5.2.2. Idle & Constant rpm Accelerometer Results

Next, idle and constant rpm data from the MALTESE collection was processed. The end-to-end process was performed on 12,420 samples, 4,140 samples per class, of idle and constant rpm data. Table 3 shows the confusion matrices for each of the four classifiers processed using all features and features selected by ReliefF (shown in green) and SFS (shown in red). When constant rpm was added to the idle data the classification performance degraded. The classification results of KNN-3, decision tree, and SVM only decreased a few percent. The separability of class BA severely decreased using all features in Naïve Bayes. When using all of the features, the kappa statistics were over 0.91 except for Naïve Bayes which was 0.5.

When ReliefF and SFS were selected 5 features were used from ReliefF and SFS selected 10 features. The top 5 ReliefF features selected were MFCC0, peak count, MFCC3, flux, and MFCC1. SFS selected MFCC0, MFCC3, MFCC9, LPC2, LPC3, RMS, peak count, dominant frequency 1, dominant frequency 2, and flux. The performances of KNN-3, decision trees, and SVM decreased minimally when using the ReliefF and SFS selected features. The features selected by SFS performed better than those selected by ReliefF. When Naïve Bayes was used with the selected features, the separability of class BA improved significantly. The kappa statistics for KNN-3 and decision trees remained above 0.96 when using both feature selectors. SVM and Naïve Bayes on the other hand had kappa statistics between 0.64 and 0.72 when ReliefF and SFS were utilized.

Table 3. Confusion matrices from using idle and constant rpm MALTESE data.

Naïve Bayes

All Feature	AA	BA	CA
AA	96.81	2.44	0.75
BA	34.73	33.50	31.76
CA	12.95	4.57	82.49

Naïve Bayes

ReliefF	AA	BA	CA
AA	87.32	10.99	1.69
BA	11.81	67.34	20.85
CA	7.90	17.87	74.23

Naïve Bayes

SFS	AA	BA	CA
AA	92.77	6.67	0.56
BA	17.17	51.55	31.28
CA	0.36	5.39	94.25

KNN-3

All Feature	AA	BA	CA
AA	100	0	0
BA	0.10	99.90	0
CA	0	0	100

KNN-3

ReliefF	AA	BA	CA
AA	99.32	0.32	0.36
BA	0.26	98.70	1.04
CA	0.41	1.14	98.45

KNN-3

SFS	AA	BA	CA
AA	99.88	0.12	0
BA	0.29	99.69	0.02
CA	0	0.29	99.71

Decision Trees

All Feature	AA	BA	CA
AA	99.42	0.58	0
BA	0.48	98.72	0.80
CA	0.07	0.56	99.37

Decision Trees

ReliefF	AA	BA	CA
AA	98.86	0.87	0.27
BA	1.04	96.98	1.98
CA	0.31	2.34	97.34

Decision Trees

SFS	AA	BA	CA
AA	98.92	1.01	0.07
BA	1.30	97.32	1.38
CA	0.22	1.57	98.21

SVM

All Feature	AA	BA	CA
AA	96.67	3.31	0.02
BA	4.11	92.92	2.97
CA	0.24	5.73	94.03

SVM

ReliefF	AA	BA	CA
AA	85.75	11.40	2.85
BA	8.48	79.44	12.08
CA	8.09	19.52	72.39

SVM

SFS	AA	BA	CA
AA	88.24	11.47	0.29
BA	7.68	75.43	16.90
CA	3.58	14.57	81.85

5.2.3. Piston vs. Turbine Accelerometer Results

The last test using only the MALTESE set of data was done to demonstrate the separability of targets with piston and turbine engines. The sample set of idle and constant rpm data was used for this test. Since vehicles AA and BA, had piston engines, 2,070 samples from each vehicle were used. All 4,140 samples from vehicle CA, which had a turbine engine, were used. Near perfect classification occurred for KNN-3, decision trees, and SVM when all features were used. Naïve Bayes was also successful with

classification rates higher than 74%. All of the kappa statistics were considered successful, higher than 0.9, except for Naïve Bayes. Table 4 contains the confusion matrices for piston vs. turbine classification using all features. ROC curves for decision tree classification using all features are also shown in Figure 17. The area under the ROC curve (AOC) was 0.995. An AOC higher than 0.91 were seen for the classifiers using all features.

Table 4. Piston vs. Turbine MALTESE confusion matrices from using all features.

Naïve Bayes	P	T
P	74.11	25.87
T	6.30	93.70

KNN-3	P	T
P	100	0
T	0	100

Decision Trees	P	T
P	99.25	0.75
T	0.92	99.08

SVM	P	T
P	94.52	5.48
T	4.20	95.80

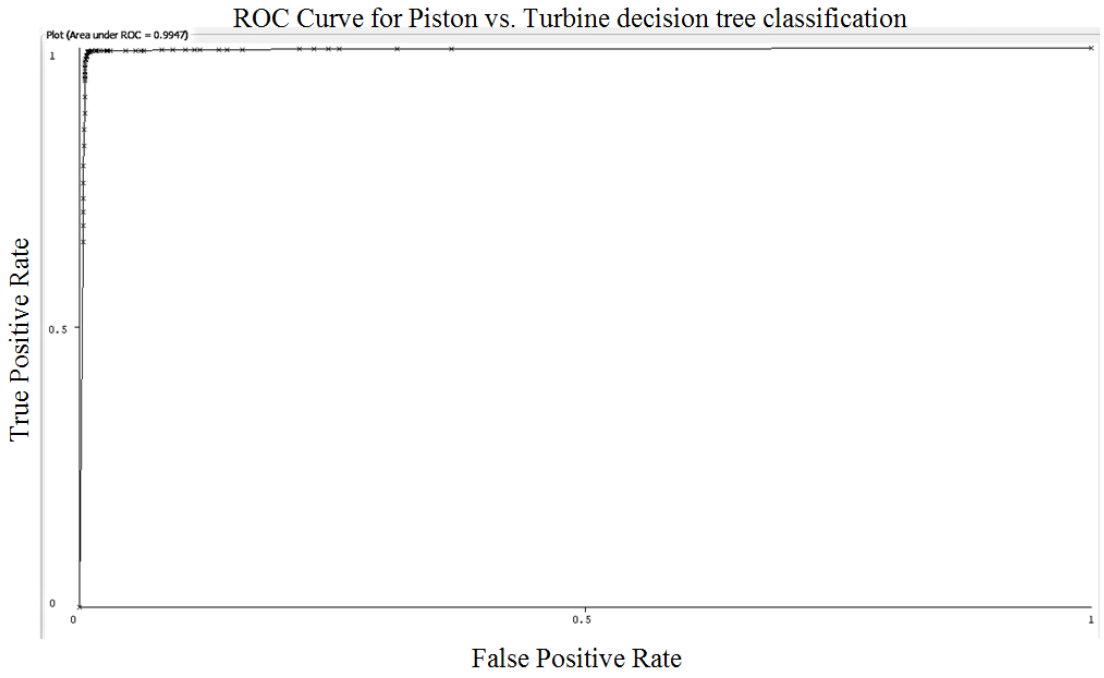


Figure 17. ROC curve generated by WEKA for piston vs. turbine decision tree classification with AOC=0.99.

Five ReliefF features and ten SFS features were selected to classify the piston vs. turbine data. The 10 SFS features selected were MFCC0, MFCC2, MFCC3, LPC1, LPC3, LPC8, LPC9, RMS, peak count, and dominant frequency 1. The ReliefF features selected were MFCC0, MFCC1, MFCC2, MFCC3, and peak count. Slight degradation was seen overall in the classification rates when only using the features selected by ReliefF and SFS. The confusion matrices for SFS and ReliefF features are in Appendix A. The features selected by SFS slightly outperformed the features selected by ReliefF.

5.3. LAVA-SC Results

Data from LAVA-SC was collected and used to analyze the end-to-end process using simultaneously collected laser vibrometer and accelerometer. The end-to-end process was tested with five different configurations of LAVA-SC data: idle accelerometer data, idle and constant rpm accelerometer data, idle laser vibrometer data, train with accelerometer

data and test with laser vibrometer data, and train on idle accelerometer data and test with constant rpm accelerometer data. The first two cased used accelerometer data only, where idle accelerometer data only was tested first and followed by a combination of idle and constant rpm accelerometer data. The end-to-end process was tested next using idle laser vibrometer data. A combination of accelerometer and laser vibrometer data was classified using accelerometer data to train the classifier and then perform classification tests using laser vibrometer data. Finally, the robustness of the features was tested by training the classifiers on idle and testing them on constant rpm data and vice versa.

5.3.1. Idle Accelerometer Results

The first test of the end-to-end process was completed using idle accelerometer data from LAVA-SC. The process was completed with 9,000 samples, with 3,000 samples per class of idle data. The confusion matrices for all four classifiers using all features and the feature selected by Relief (in green) and SFS (in red) are given in Table 5. When all features were used for classification, KNN-3, decision trees, and SVM performed successfully with classification rates of 97% and higher. The classification rates for Naïve Bayes were much worse, between 58% and 69%, when all features were used. Naïve Bayes, however, improved significantly when features selected by ReliefF and SFS were used. The top 5 ReliefF features selected were MFCC0, MFCC1, MFCC2, MFCC3, and peak count. SFS selected 5 features: MFCC0, LPC7, dominant frequency 1, dominant frequency 2, and flux. The performance of KNN-3, decision trees, and SVM decreased when only using the five features selected by each of the feature selection techniques. The performance of SVM degraded significantly when the features selected by ReliefF were used. Only KNN-3 and decision trees received kappa statistics higher

than 0.9 regardless of the features used. When all features were used, SVM had a successful kappa statistic; however, the value decreased significantly when ReliefF and SFS were used. This degradation could be due to removing features that helped to maximize the margin between the hyperplanes formed to separate the classes in SVM.

Table 5. Idle accelerometer LAVA-SC confusion matrix results.

Naïve Bayes

All Feature	D	H	C
A	58.84	26.23	14.93
B	1.67	58.83	12.70
C	6.30	24.63	69.07

Naïve Bayes

ReliefF	D	H	C
D	69.20	21.17	9.63
H	8.17	84.73	7.10
C	20.23	34.13	45.64

Naïve Bayes

SFS	D	H	C
D	81.37	11.33	7.30
H	9.80	81.63	8.57
C	24.67	14.83	60.50

KNN-3

All Feature	D	H	C
A	99.94	0.03	0.03
B	0.03	99.9	0.07
C	0	0	100

KNN-3

ReliefF	D	H	C
D	99.10	0.57	0.33
H	0.87	98.27	0.87
C	0.17	0.30	99.53

KNN-3

SFS	D	H	C
D	99.03	0.27	0.70
H	0.63	94.67	4.70
C	0.40	3.97	95.63

Decision Tree

All Feature	D	H	C
A	98.80	0.53	0.67
B	0.47	98.53	1
C	0.60	0.63	98.77

Decision Tree

ReliefF	D	H	C
D	96.87	1.40	1.73
H	1.77	96.83	1.40
C	1.77	1.60	96.63

Decision Tree

SFS	D	H	C
D	98.90	0.43	0.67
H	0.27	96.13	3.60
C	1.30	4.10	94.60

SVM

All Feature	D	H	C
A	97.67	0	2.33
B	0.20	98.83	0.97
C	2.17	0.20	97.63

SVM

ReliefF	D	H	C
D	62.07	5.77	32.16
H	2.80	74.40	22.80
C	24.93	8.57	66.50

SVM

SFS	D	H	C
D	84.74	6.03	9.23
H	10.50	81.70	7.80
C	16.40	13.77	69.83

5.3.2. Idle & Constant rpm Accelerometer Results

Idle and constant rpm accelerometer data were used in the end-to-end process next. A total of 18,000 samples, 6,000 samples per class, of idle and constant rpm data were tested. Three of the four classifiers performed successfully when all features were used. Naïve Bayes performed poorly at correctly classifying class D when all features were used. Table 6 displays the confusion matrices for the classifiers when all features were used. The rest of the confusion matrices using idle and constant rpm data can be seen in Appendix A. The 5 features selected by ReliefF were MFCC0, MFCC2, MFCC5, MFCC8, and spectral centroid. MFCC0, MFCC2, MFCC5, MFCC7, MFCC9, dominant frequency 1, dominant frequency 2, and flux were the 8 features selected by SFS. Slight improvements were seen in the overall classification when Naïve Bayes was used with the ReliefF and SFS features. The performance of SVM decreased severely when only the features selected by SFS and ReliefF were used.

Table 6. Idle and constant rpm accelerometer LAVA-SC results using all features.

Naïve Bayes	D	H	C
D	43.68	40.12	16.20
H	2.37	83.13	14.50
C	3.65	13.23	83.12

KNN-3	D	H	C
D	100	0	0
H	0.05	99.87	0.08
C	0	0	100

Decision Trees	D	H	C
D	98.48	1.15	0.37
H	1	98	1
C	0.32	0.70	98.98

SVM	D	H	C
D	96.98	0.58	2.42
H	0.72	98.30	0.98
C	2.55	0.75	96.70

5.3.3. Idle Laser Vibrometer Results

Analysis of the idle laser vibrometer data was completed using the end-to-end process. The four classifiers were processed with 1,491 samples of data, 497 samples per class. The overall performance of the classifiers using all features was successful. All of the classifiers but Naïve Bayes classified with greater than 95% accuracy. Confusion matrices for the classifiers using all features are shown in Table 7. When feature selection was used, 10 features were selected by SFS. The 10 features selected were MFCC0, MFCC2, MFCC4, MFCC5, MFCC7, MFCC8, zero crossings, dominant frequency 1, and flux. The top 5 ReliefF features were MFCC0, MFCC2, MFCC4, MFCC7, and rolloff. The classification rates of Naïve Bayes increased to an average of 89% when the features selected by SFS were used. The kappa statistic remained below 0.8 even with an improved classification rate. The remaining classifiers performed slightly worse using ReliefF and SFS than when all features were used. The kappa statistics remained above 0.95 for KNN-3, decision trees, and SVM regardless of the features.

Table 7. Idle laser vibrometer LAVA-SC confusion matrices all features.

Naïve Bayes	D	H	C
D	86.08	11.96	1.96
H	9.24	86.86	3.90
C	11.54	9.51	78.95

KNN-3	D	H	C
D	100	0	0
H	0.21	99.79	0
C	0	0	100

Decision Trees	D	H	C
D	98.82	0.78	0.40
H	0.62	95.28	4.11
C	0.20	2.63	97.17

SVM	D	H	C
D	100	0	0
H	0.41	99.18	0.41
C	1.42	0	98.58

5.3.4. Train Accelerometer Data/ Test Laser Vibrometry Data Results

After analyzing accelerometer and laser vibrometry data separately, a combination of the two sensors was tested. Each classifier was trained using 2,037 samples, 679 samples per class, of idle and constant rpm accelerometer data and tested using 2,037 samples of idle and constant rpm laser vibrometer data. The process was tested without feature normalization and with feature normalization, or normalizing each feature from 0 to 1. Normalizing each of the features was tested to remove extra scaling differences between the accelerometer data and laser vibrometry data. Without feature normalization, all of the classifiers performed poorly. Naive Bayes classified everything as target C. SVM classified everything as target H. KNN-3 and decision trees performed badly when trying to classify class D but performed reasonably for the other two classes. When each of the features were normalized the overall performance of the classifiers improved. Table 8 contains the confusion matrices for training with accelerometer data and testing on laser vibrometer data with and without feature normalization.

Table 8. Training on accelerometer data and testing on laser vibrometer data without/ with feature norm.

Without Normalization

Naïve Bayes	D	H	C
D	0	2.06	97.94
H	0	16.94	83.06
C	0	4.12	95.88

KNN-3	D	H	C
D	22.09	41.09	36.82
H	8.98	75.85	15.17
C	5.30	22.39	72.31

Decision Trees	D	H	C
D	13.40	59.20	27.40
H	11.93	60.09	27.98
C	0.88	48.60	50.52

SVM	D	H	C
D	1.18	98.82	0
H	0	100	0
C	0	100	0

With Normalization

Naïve Bayes	D	H	C
D	61.86	5.45	32.70
H	10.75	46.98	42.27
C	4.13	9.57	86.30

KNN-3	D	H	C
D	82.33	1.03	16.64
H	16.79	68.34	14.75
C	14.14	3.68	82.18

Decision Trees	D	H	C
D	93.09	12.22	8.39
H	33.14	53.46	13.40
C	18.56	16.20	65.24

SVM	D	H	C
D	95.14	4.12	0.74
H	30.78	67.89	1.33
C	39.76	6.48	53.76

5.3.5. Train Idle Data/ Test Constant rpm Data & Vice Versa Results

The final test done using LAVA-SC data was training on one engine condition and testing on another. A set of 9,000 idle samples and 9,000 constant rpm samples were processed using each classifier. The end-to-end process was tested by training the classifiers using idle accelerometer data and testing on constant rpm accelerometer data and vice versa. The confusion matrices for each of these methods are shown in Table 9.

Both methods of testing performed poorly. When training the classifiers with idle data and testing with constant rpm data, class C performed well within each of the classifiers. When the reverse was tested, however, class H performed well within each of the classifiers instead. A lack of features that were robust across engine conditions was one explanation to the performance changes.

Table 9. Training on idle data and testing on constant rpm data and vice versa.

Train Idle/ Test Constant rpm

Naïve Bayes	D	H	C
D	34.30	16.30	49.4
H	16.73	31.17	52.10
C	21.20	0	78.80

KNN-3	D	H	C
D	64.6	18.17	17.23
H	37.60	43.63	18.77
C	2.53	2.23	95.24

Decision Trees	D	H	C
D	41.20	7.90	50.90
H	15.1	41.37	43.53
C	4.43	6.27	89.30

SVM	D	H	C
D	92.70	2.33	4.97
H	37.50	46.77	15.73
C	9.23	0.07	90.70

Train Constant rpm/ Test Idle

Naïve Bayes	D	H	C
D	38.33	49	12.67
H	10.50	86.53	2.97
C	0.33	46.47	53.20

KNN-3	D	H	C
D	77.70	9.87	12.43
H	4.90	94.50	0.60
C	14.77	40.70	44.53

Decision Trees	D	H	C
D	50	26.27	23.73
H	3.03	96.53	0.44
C	10.77	49.07	40.17

SVM	D	H	C
D	83.33	3.40	13.27
H	17.17	82.57	0.27
C	23.03	34.63	42.34

5.4. Combined Results

MALTESE and LAVA-SC data was combined to analyze a larger class dataset. Three different combination cases were considered: idle accelerometer data, idle and constant rpm data, and a military vehicle vs. civilian vehicle comparison. One analysis was done using only idle accelerometer data from MALTESE and LAVA-SC. The second analysis was done using idle and constant rpm accelerometer data. The final analysis combined all of LAVA-SC into a single class called civilian and MALTESE into a single class called military to compare the separability between military and civilian vehicles.

5.4.1. 6- Class Idle Accelerometer Data Results

Idle data from MALTESE and LAVA-SC was processed first. Each class was represented with 720 samples, 4,320 total samples. Classification with all features was successful for each of the classifiers. Naive Bayes performed the worst of the four classifiers when all features were used. The confidence in each classification was high with each classifier having a kappa statistic greater than 0.84. Table 10 contains the confusion matrices for each of the classifiers when all results were used. When SFS was used to select features 13 features were selected: MFCC0, MFCC1, MFCC3, MFCC4, MFCC8, LPC8, zero crossing, RMS, peak count, dominant frequency 1, dominant frequency 2, dominant frequency 3, and flux. The top 5 ReliefF features used were spectral centroid, MFCC0, MFCC1, MFCC3, and peak count. The classification rates decreased slightly when ReliefF and SFS features were used. Naïve Bayes performed the worst when only the ReliefF features were used. The features selected by SFS performed better than Relief's features. The classification results for each of the feature selection methods are shown in Appendix A.

Table 10. Combined 6-class MALTESE and LAVA-SC confusion matrix results for idle data using all features.

Naïve Bayes	AA	BA	CA	D	H	C
AA	98.75	1.25	0	0	0	0
BA	16.67	81.94	1.39	0	0	0
CA	0	0.14	99.86	0	0	0
D	1.11	6.11	0	83.89	7.36	1.53
H	1.11	0	0	2.78	92.22	3.89
C	0.14	4.03	0	8.75	18.89	68.19

KNN-3	AA	BA	CA	D	H	C
AA	100	0	0	0	0	0
BA	0	100	0	0	0	0
CA	0	0	100	0	0	0
D	0	0	0	100	0	0
H	0	0	0	0.14	99.86	0
C	0	0	0	0	0	100

Decision Tree	AA	BA	CA	D	H	C
AA	99.44	0.28	0	0	0	0.28
BA	0.28	98.88	0.42	0.42	0	0
CA	0.14	0.42	99.30	0.14	0	0
D	0.28	0.14	0.14	97.50	1.11	0.83
H	0.14	0	0	0.97	95.83	3.06
C	0	0	0	0.42	2.50	97.08

SVM	AA	BA	CA	D	H	C
AA	100	0	0	0	0	0
BA	0	100	0	0	0	0
CA	0	0	100	0	0	0
D	0	0	0	100	0	0
H	0	0	0	0	98.19	1.81
C	0	0	0	0.42	0.14	99.44

5.4.2. 6-Class Idle and Constant rpm Accelerometer Data Results

Next, idle and constant rpm accelerometer data from MALTESE and LAVA-SC was tested in the end-to-end process. A total of 4074 samples, 679 samples per class, were processed for a 6-class classification. KNN-3, decision trees, and SVM successfully classified the data when all features were used with average classification rates 93% and higher. KNN-3 and decision trees performed well when feature selection was used as well. The performance of KNN-3 and decision trees degraded slighted when features were selected by SFS and ReliefF. The top 5 ReliefF features were RMS, MFCC0, MFCC3, spectral centroid, and flux. The 12 features SFS selected were MFCC0, MFCC5, MFCC9, LPC1, LPC2, rolloff, RMS, peak count, dominant frequency 1,

dominant frequency 2, spectral ratio 3, and flux. The results for all of the confusion matrices are displayed in Table 11 and Appendix A.

Table 11. Combined 6-class results for idle and constant rpm data using all features.

Naïve Bayes	AA	BA	CA	D	H	C
AA	96.38	3.19	0.43	0	0	0
BA	35.51	28.76	32.03	0	0	0
CA	13.24	4.26	82.50	0	0	0
D	0	0	0	86.60	4.27	9.13
H	0	0	0	13.84	77.91	8.25
C	0	0	0	32.99	14.01	53

KNN-3	AA	BA	CA	D	H	C
AA	99.86	0.14	0	0	0	0
BA	0	100	0	0	0	0
CA	0	0.46	99.54	0	0	0
D	0	0	0	100	0	0
H	0	0	0	0	100	0
C	0	0	0	0	0	100

Decision Tree	AA	BA	CA	D	H	C
AA	97.68	1.59	0.43	0	0	0
BA	2.46	95.22	2.32	0	0	0
CA	0.46	2.44	97.10	0	0	0
D	0	0	0	97.05	1.18	1.77
H	0	0	0	1.60	95.60	2.80
C	0	0	0	1.32	3.68	95

SVM	AA	BA	CA	D	H	C
AA	95.28	5.74	0.59	0	0	0
BA	4.57	93.37	3.68	0	0	0
CA	0.15	9.28	87.33	0	0	0
D	0	0	0	93.37	0.29	6.33
H	0	0	0	1.47	97.20	1.32
C	0	0	0	4.42	0.15	95.43

5.4.3. Military vs. Civilian Accelerometer Data Results

The final test done using MALTESE and LAVA-SC was to separate military and civilian vehicles. The same data from the last test, 4,074 samples of idle and constant rpm data, was used to classify the vehicles. Classes AA, BA, and CA were combined to create the military vehicle class M. Classes D, H, and C were combined to create civilian class C. Near perfect classification was seen from all classifiers when all features were used. The performance of Naïve Bayes improved when ReliefF and SFS features were used. The top 5 ReliefF features were RMS, rolloff, spectral centroid, MFCC0, and zero crossing. The 7 features SFS selected were MFCC0, MFCC4, MFCC8, MFCC9, RMS, peak count, and zero crossing. Confusion matrices for all of the results are displayed in

Table 12 and Appendix A. A ROC curve, shown in Figure 18, illustrates the performance of Naïve Bayes when all features were used. This ROC curve was the worst performance seen for military vs. civilian vehicle classification. The area under the curve (AOC) for Naïve Bayes was 0.99. The remainder of the classification results had AOCs even closer to one.

Table 12. Military vs. civilian results using idle and constant rpm data with all features.

Naïve Bayes	M	C
M	98.38	1.62
C	0	100

KNN-3	M	C
M	100	0
C	0	100

Decision Trees	M	C
M	99.96	0.04
C	0	100

SVM	M	C
M	100	0
C	0	100

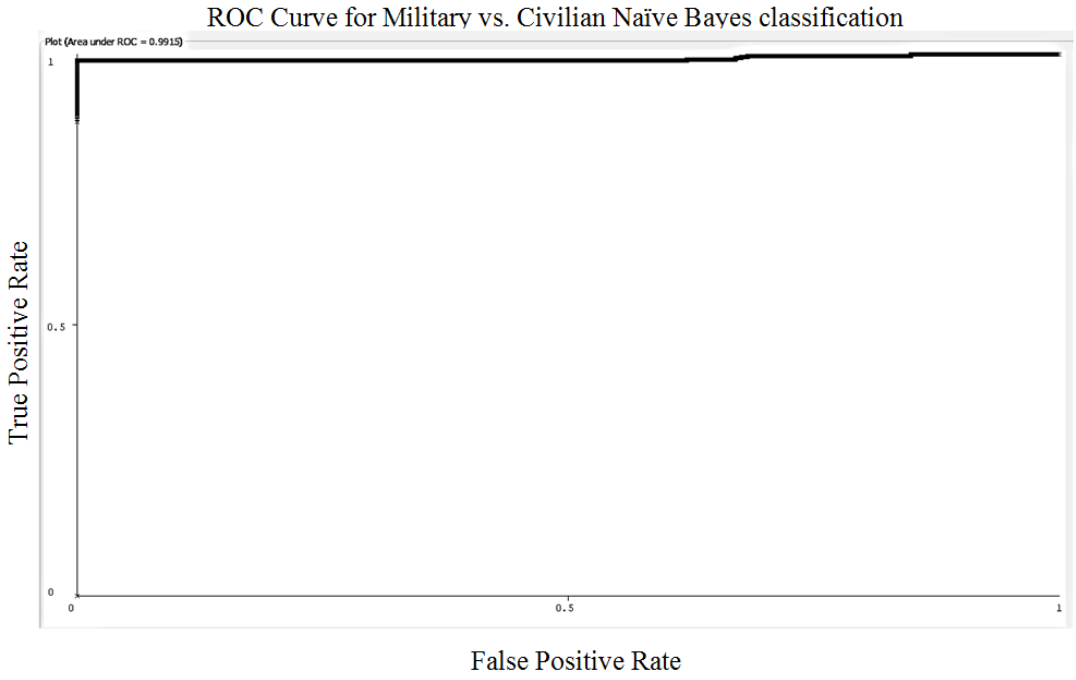


Figure 18. ROC curve generated by WEKA for military and civilian Naïve Bayes classification with AOC=0.99.

5.5. Performance Summary

The end-to-end process was tested with 11 different combinations of accelerometer and laser vibrometer data. A total of 3 tests were performed on MALTESE data, 5 tests on LAVA-SC data, and 3 tests on a combination of the two datasets. Each of these tests was performed to examine the robustness of the end-to-end process. After each test, the performances of the features, feature selectors, and classifiers were examined to demonstrate the ability of the process to determine the best methods to exploit vibrometry data.

The extracted features were examined to find features that could perform robustly for different datasets and different operating conditions. Some of the features that were selected repeatedly in various cases included MFCC0, MFCC3, peak count, dominant frequency 1, and flux. It may be emphasized that the first-order statistics suggested that

these features would be useful for distinguishing between MALTESE and LAVA-SC vehicles. The only feature selected by ReliefF and SFS in every experiment was MFCC0. In most cases, using the features selected by ReliefF and SFS provided classifications close to the performance when all features were used. In some cases, Naïve Bayes improved classification when fewer features were selected. The features selected by SFS often outperformed those selected by ReliefF. SFS, however, often selected double the amount of features used for classification. Overall, the classification performances generally exceeded 85% successful classification. KNN-3 and decision trees generally performed the best among the four classifiers. The few cases that did not classify the vehicles as well were often when Naïve Bayes was used, when training on one set of data and testing on another, and when SVM was performed using ReliefF and SFS features.

6. CONCLUSIONS

This thesis develops an end-to-end classification framework for effective identification of vehicles using vibrometry data. The velocity of vibrating surfaces of a vehicle creates a unique signature which can be exploited for vehicle identification. These signatures can be enhanced by mapping the raw velocity data into statistical features. The classification process developed to exploit the vibration signatures involves preprocessing the data, extracting 11 features, performing two feature selection methods, and classifying the features with four classifiers. The baseline end-to-end framework largely follows previous works, including [1], [2], and [5], and is implemented using MATLAB and WEKA. Preprocessing and feature extraction were performed using MATLAB. WEKA was used for feature selection, classification, and analysis of the results.

The features selected for this end-to-end process were chosen because they are well established in speech processing, seismology, and structural analysis [1]. Investigating the meaning and purpose of each feature revealed overlap and redundancies as well as application differences for the information collected. For example, zero crossings, spectral centroid, and dominant frequencies are all related to each other in the sense that one set of features can be used to calculate another. In spite of the redundancy among some of the features, the overall performance improvement is significant when classification is performed using features in comparison to classification performance

using raw time series data. It is expected that development of features optimized for remote vibrometry and the expected targets will also significantly increase performance.

The feature selection techniques used in the end-to-end process were ReliefF [27] and SFS [26]. ReliefF was previously used in [1] and [2] to select vibrometry features. SFS was chosen as a baseline feature selection technique. Overall, similar performances were seen from classifying with features selected by ReliefF and SFS. SFS often outperformed ReliefF, but SFS used nearly double the number of features attain the moderately improved performance. SFS selected between 5 and 13 features for each experiment; whereas the number of features for ReliefF was restricted to 5 features to create results similar to [1]. The only feature that was consistently selected for use by the each feature selector in every test was MFCC0. The lack of consistent selection suggests a need for more robust features for the exploitation of vibrometry data. Using features based on vehicles phenomenology, like piston firing rate, axle count, and engine configuration, could provide a more robust classification of vehicles.

Lastly, the end-to-end process was successful at identifying the vehicles from MALTESE and LAVA-SC. Among the four classifiers, KNN-3 and decision trees classified most the vehicles with classification rates of 95% and higher. High confidence was also seen in these classifiers with kappa statistics often greater than 0.85. SVM classified the vehicles successfully as well, but performance severely degraded in some cases when fewer features were used. Naïve Bayes performed the worst among the four classifiers for many of the experiments. Naïve Bayes, however, was the only classifier that improved when fewer features were used. Out of all of the tests, the military vs. civilian and piston vs. turbine experiments performed the best using all classifiers. When

trying to separate military vehicles from civilian vehicles and piston engine vehicles from turbine engine vehicles the classifiers were highly effective even though less detail about the vehicles was utilized. The classification rates of military vs. civilian vehicles were 98% and higher. Piston vs. turbine classification rates were also 95%+ with high confidence in all classifiers except for Naïve Bayes.

The overall development and performance of the end-to-end process to aid development of an AiTR process was highly successful for MALTESE and LAVA-SC datasets. Improvements to the end-to-end process can continue to be made for further enhancement of classification rates and classifier confidences. Using other features, state of art the feature selection and classification algorithms, and collecting data with more operating conditions are among several ways this end-to-end process can be improved.

7. FUTURE WORK

Since using vibrations as a vehicle identifier is relatively unexplored, many possibilities exist in the future of this research. Improving the end-to-end process, data fusion with synthetic aperture radar (SAR) or other modalities, investigating vibration stability across the vehicle, and hierarchical classification are being considered for future projects. These ideas are just a few possibilities of what can take place for future research topics on the exploitation of laser vibrometry data.

Many improvements can still be made to this end-to-end process. As previously stated the features selected by Stevens in [1] were chosen due to their effectiveness in other fields and because the vibration data is relatively similar to speech and other data. Selecting, developing, and testing new or other features that are more applicable to vibrometry needs completed to best exploit vibrometry data. Features based on the vehicles axle count, engine configuration, and axle count could be used to exploit vibrometry data. Furthermore, testing new feature selection and classification methods will provide additional insight into feature utility. Determining the effects of sensor, beam location, angle, and beam size on a vehicle's vibration signature may also allow for a more robust classification process.

Fusing laser vibrometry with SAR images of targets may provide a novel way to identify confused targets. AFRL has a database of SAR images called Moving and Stationary Target Acquisition and Recognition (MSTAR). There is an overlap of 17

vehicles among the MSTAR data and the existing database of vibrometry data. Based on initial analysis at the scoring metrics used to classify the SAR images, 12 of the 17 vehicles are not easily identified. Using vibration data as a complement to the SAR data may provide an opportunity to improve separation of vehicles that are typically confused in the SAR-only domain. The improvement in separability would be due to the fact that two vehicles could look very similar from a distance, but may have different vibration signatures.

Finally, future work can also be done to create a hierarchical classifier to identify vehicles. Hierarchical classification allows for decisions about broad classes vehicles to be made in steps such as military vehicles vs. civilian vehicles, piston engines versus turbine engines, small vehicle vs. large vehicles, and others. Classification on military vs. civilian and piston vs. turbine for a small dataset has been shown to be successful in this thesis. Expanding on the success and incorporating them into an overall hierarchy requires further research. Understanding how to organize the decision levels, how to pass information between levels, and how to pass information from one branch of the hierarchy to the other are all questions that can be explored to implement hierarchical classification for vibrometry.

REFERENCES

[1] M. R. Stevens, M. Snorrason and D. Petrovich, "Laser vibrometry for target classification," in *Proceedings of SPIE 4726*, 2002.

[2] M. R. Stevens, D. W. Stouch, M. Snorrason and F. Heitkamp, "Mining vibrometry signatures to determine target separability," in *SPIE Automatic Target Recognition XIII*, 2003.

[3] A. Nooralahiyan, L. Lopez, D. Mckewon and M. Ahmad, "Time-delay neural network for audio monitoring of road traffic and vehicle classification," in *Proceedings of SPIE 2902*, 1997.

[4] V. Masagutov, D. W. Stouch, P. Kanjilal and M. Snorrason, "Vibrometry classification of moving vehicles using throttle signature analysis," in *IEEE International Conference on Systems, Man and Cybernetics*, 2007.

[5] S. Kangas, O. Mendoza-Schrock and A. Freeman, "Applying manifold learning to vehicle classification using vibrometry signatures," in *Proceedings of SPIE 8751*, 2013.

[6] W. Ma, D. Xing, A. McKee, R. Bajwa, C. Flores, B. Fuller and P. Varaiya, "A wireless accelerometer-based automatic vehicle classification prototype system," *IEEE Transactions on Intelligent Transportation System,* vol. 15, no. 1, pp. 104-111, 2014.

[7] L. Crider and S. Kangas, "Exploiting vibration-based spectral signatures for automatic target recognition," in *SPIE 9079, Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR V*, 2014.

[8] I. Nedgard, "A Comparison of Analysis Methods for Vehicle Classification by Laser Vibrometry," Swedish Defense Research Agency, Stockholm, 2005.

[9] "Polytec: Basic Principles of Vibrometry," Polytec GmbH Waldbronn, 2014. [Online]. Available: http://www.polytec.com/us/solutions/vibration-measurement/basic-principles-of-vibrometry/. [Accessed 10 March 2014].

[10] D. Jameson, "Effects of spatial modes on ladar vibration signature estimation," University of Dayton, 2007.

[11] B. Bhanu, "Automatic target recognition: state of the art survey," *IEEE Transactions on Aerospace and Electronic Systems,* vol. 22, no. 4, pp. 364-379, 1986.

[12] L. M. Novak, G. J. Owirka, W. S. Brower and A. L. Weaver, "The automatic target-recognition system in SAIP," *The Lincoln Laboratory Journal,* vol. 10, no. 2, pp. 187-202, 1997.

[13] A. Smith, O. Mendoza-Schrock, S. Kangas, M. Dierking and A. Shaw, "An end-to-end vehicle classification pipeline using vibrometry data," in *SPIE 9079, Ground/Air Multisensor Interoperability, Integration, and Networking for Persistent ISR V* , 2014.

[14] S. Theodoridis and K. Koutroumbas, Pattern Recognition (Chapter 6), Academic Press, 2008.

[15] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *IEEE, Acoustics, Speech and Signal Processing 2*, 1997.

[16] B. Arons, "SpeechSkimmer: a system for interactively skimming recorded speech," *ACM Transactions on Computer-Human Interaction,* vol. 4, no. 1, pp. 3-38, 1997.

[17] L. R. Rabiner and M. R. Sambur, "An algorithm for determining the endpoints of isolated utterances," *The Bell System Technical Journal,* vol. 54, no. 2, pp. 297-315, 1975.

[18] B. Gajíc and K. Paliwal, "Robust speech recognition using features based on zero crossings with peak amplitudes," in *IEEE, Acoustics, Speech and Signal Processing 1*, 2003.

[19] P. A. Sobel and D. H. von Seggern, "Analysis of Selected Seismic Events from Asia in a Seismic Discrimination Context," Teledyne Geotech (SDAC-TR-78-5),

Alexandria, VA, 1978.

[20] T. Tiira, "Discrimination of nuclear explosions and earthquakes from teleseismic distances with a local network of short period seismic stations using artificial neural networks," *Physics of the Earth and Planetary Interiors,* vol. 97, pp. 247-268, 1996.

[21] G. Tzanetakis and P. Cook, "A framework for audio analysis based on classification and temporal segmentation," in *EUROMICRO 25*, 1999.

[22] J. Makhoul, "Linear prediction: a tutorial review," in *IEEE 63(4),* 1975.

[23] L. Muda, M. Begam and I. Elamvazuthi, "Voice recognition algorithms using Mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques," *Journal of Computing* , vol. 2, no. 3, pp. 138-143, 2010.

[24] H. Kim and W. H. Holmes, "Nonparametric peak feature extraction and its applications to speech signals," in *8th Australian International Conference of Speech Science and Technology,* 2000.

[25] O. Lartillot and P. Toiviainen, "A Matlab toolbox for musical feature extraction from audio," in *10th International Conference on Digital Audio Effects*, 2007.

[26] S. F. Pramata, A. K. Muda, Y. H. Choo and N. A. Muda, "Computationally inexpensive sequential forward floating selection for acquiring significant features for authorship invarianceness in writer identification," *International Journal on New Computer Architectures and Their Applications*, pp. 581-598, 2011.

[27] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Machine Learning Journal,* vol. 53, pp. 23-69, 2003.

[28] A. W. Whitney, "A Direct Nonparametric Measurement Selection," in *IEEE Transactions on Computers*, 1971.

[29] K. Kira and L. Rendell, "The feature selection problem: traditional methods and a new algorithm," in *Tenth National Conference on Artificial Intelligence*, 1992.

[30] A. J. Viera and J. M. Garrett, "Understanding interobserver agreement: the kappa statistic," *Family Medicine,* vol. 37, no. 5, pp. 360-363, 2005.

[31] T. M. Mitchell, Machine Learning (Draft), 2010, pp. 1-17.

[32] O. Sutton, "Introduction to k nearest neighbour classification and condensed nearest

neighbor data reduction," 2012.

[33] J. R. Quinlan, "Improved use of continuous attributes in C4. 5," *Journal of Artificial Intelligence Research* , vol. 4, pp. 77-90, 1996.

[34] J. R. Quinlan, C4.5 Programs for machine learning, London: Morgan Kaufmann Publishers Inc., 1988.

[35] D. Eads, D. Hill, S. Davis, S. Perkins, J. Ma, R. Porter and J. Theiler, "Genetic algorithms and support vector machines for times series classification," in *SPIE 4787*, 2002.

[36] C. W. Hsu, C. C. Chang and C. J. Lin, "A practical guide to support vector classification," 2010. [Online]. Available: http://www.csie. ntu.edu.tw/~cjlin. [Accessed 15 January 2014].

[37] J. C. Platt, "Sequential minimal optimization: a fast algorithm for training support vector machines," Microsoft Research (MSR-TR-98-14), 1998.

[38] C. J. Burges, "A tutorial on support vector machines for pattern recognition," in *Data Mining and Knowledge Discovery*, Boston, Kluwer Academic Publishers, 1998, p. 121–167.

[39] I. W. Tsang, J. T. Kwok and P. M. Cheung, "Very Large SVM Training using Core Vector Machines".

APPENDIX A: Additional Results

Table 13. Piston vs. Turbine SFS classification results.

Naïve Bayes	T	P
T	92.5	7.46
P	18.27	81.73

KNN-3	T	P
T	99.83	0.17
P	0.05	99.95

Decision Trees	T	P
T	98.41	1.59
P	1.55	98.45

SVM	T	P
T	91.21	8.79
P	17.18	82.82

Table 14. Piston vs. Turbine ReliefF classification results.

Naïve Bayes	T	P
T	81.23	18.77
P	20.13	79.87

KNN-3	T	P
T	99.57	0.43
P	0.43	99.57

Decision Trees	T	P
T	97.80	2.20
P	2.20	97.80

SVM	T	P
T	80.77	19.23
P	20.68	79.32

Table 15. Results from SFS using LAVA-SC idle and constant rpm accelerometer data.

Naïve Bayes	D	H	C
D	81.77	8.05	10.18
H	17.68	73.72	8.60
C	38.83	13.32	47.85

KNN-3	D	H	C
D	99.42	0.17	0.41
H	1.34	97.78	0.88
C	0.12	0.23	99.65

Decision Trees	D	H	C
D	97.50	1.60	0.90
H	1.45	97.15	1.40
C	0.40	1.25	98.35

SVM	D	H	C
D	61.95	8.02	30.03
H	8.48	84.22	7.30
C	21.17	6.92	71.91

Table 16. Results from ReliefF using LAVA-SC idle and constant rpm accelerometer data.

Naïve Bayes	D	H	C
D	65.12	13.48	21.40
H	14.95	79.08	5.97
C	14	17.23	68.77

KNN-3	D	H	C
D	98.40	0.45	1.15
H	1.80	96.72	1.48
C	0.75	0.68	98.57

Decision Trees	D	H	C
D	95.93	2.37	1.70
H	2.95	95.28	1.77
C	1.23	1.65	97.12

SVM	D	H	C
D	66.82	10.42	22.77
H	8.45	85.20	6.35
C	14.67	7.37	77.96

Table 17. Results from SFS using LAVA-SC idle laser vibrometer data.

Naïve Bayes	D	H	C
D	96.67	2.75	0.58
H	9.86	81.72	8.42
C	6.68	2.22	91.10

KNN-3	D	H	C
D	99.80	0	0.20
H	0	99.79	0.21
C	0	0.20	99.80

Decision Trees	D	H	C
D	98.43	1.37	0.20
H	0.41	95.48	4.11
C	0.41	2.63	96.96

SVM	D	H	C
D	98.04	0	1.96
H	2.05	92.61	5.34
C	3.04	2.22	94.74

Table 18. Results from ReliefF using LAVA-SC idle laser vibrometer data.

Naïve Bayes	D	H	C
D	86.67	10.39	2.94
H	5.96	79.67	14.37
C	4.45	42.51	53.04

KNN-3	D	H	C
D	99.22	0.784	0
H	2.46	96.10	1.44
C	0	2.23	97.77

Decision Trees	D	H	C
D	98.82	0.59	0.59
H	1.44	89.94	8.62
C	0.61	7.29	92.10

SVM	D	H	C
D	95.29	0	4.71
H	0.62	81.52	17.86
C	3.64	29.76	66.60

Table 19. Results of SFS using 6-class idle accelerometer data from MALTESE AND LAVA-SC data.

Naïve Bayes	AA	BA	CA	D	H	C
AA	99.44	0.56	0	0	0	0
BA	10.82	73.9	1.25	14.03	0	0
CA	0	11.25	88.75	0	0	0
D	0	5.97	0	90.42	1.25	2.36
H	0	1.39	0	3.47	79.86	15.28
C	0	0.83	0	12.64	8.61	77.92

KNN-3	AA	BA	CA	D	H	C
AA	100	0	0	0	0	0
BA	0	100	0	0	0	0
CA	0	0	100	0	0	0
D	0	0	0	99.86	0.14	0
H	0	0	0	0.28	98.47	1.25
C	0	0	0	0.14	0.69	99.17

Decision Tree	AA	BA	CA	D	H	C
AA	99.31	0.28	0	0.41	0	0
BA	0.41	99.31	0.28	0	0	0
CA	0	0.56	99.03	0.41	0	0
D	0.42	0.28	0	96.94	0.83	1.53
H	0.14	0	0	0.97	94.58	4.31
C	0	0	0	0.97	3.75	95.28

SVM	AA	BA	CA	D	H	C
AA	100	0	0	0	0	0
BA	8.06	91.94	0	0	0	0
CA	0	0	100	0	0	0
D	0	0	0	97.08	0.28	2.64
H	0	0	0	0.14	77.92	21.94
C	0	0	0	9.72	3.19	87.09

Table 20. Results of ReliefF using 6-class idle accelerometer data from MALTESE AND LAVA-SC data.

Naïve Bayes	AA	BA	CA	D	H	C
AA	84.70	15.30	0	0	0	0
BA	9.03	87.08	3.33	0.56	0	0
CA	0	2.36	96.11	1.53	0	0
D	1.25	0.97	0	80.98	9.44	7.36
H	0	0	0	2.78	75.69	21.53
C	0	0	0	16.10	46.40	37.50

KNN-3	AA	BA	CA	D	H	C
AA	100	0	0	0	0	0
BA	0	100	0	0	0	0
CA	0	0	100	0	0	0
D	0	0	0	99.58	0.14	0.28
H	0	0	0	0.56	95.55	3.89
C	0	0	0	0.28	2.36	97.36

Decision Tree	AA	BA	CA	D	H	C
AA	99.58	0.42	0	0	0	0
BA	0.42	98.88	0.42	0.28	0	0
CA	0	0.28	99.58	0	0.14	0
D	0.14	0.56	0.14	96.67	1.11	1.38
H	0	0	0	0.56	91.39	8.05
C	0	0	0	1.11	3.75	95.14

SVM	AA	BA	CA	D	H	C
AA	91.39	8.61	0	0	0	0
BA	7.78	90.97	1.25	0	0	0
CA	0	0.83	99.17	0	0	0
D	0	0.14	0.14	92.49	3.06	4.17
H	0	0	0	3.75	65	31.25
C	0	0	0	15.28	19.30	65.42

Table 21. Results of SFS using 6-class idle and constant rpm accelerometer data.

Naïve Bayes	AA	BA	CA	D	H	C
AA	92.61	6.81	0.58	0	0	0
BA	26.23	34.49	39.28	0	0	0
CA	3.20	3.35	93.45	0	0	0
D	0	0	0	87.78	4.86	7.36
H	0	0	0	22.68	76.88	0.44
C	0	0	0	40.35	20.47	39.18

KNN-3	AA	BA	CA	D	H	C
AA	99.13	0.58	0.29	0	0	0
BA	0.87	98.55	0.58	0	0	0
CA	0	1.07	98.93	0	0	0
D	0	0	0	99.85	0.15	0
H	0	0	0	2.65	94.85	2.50
C	0	0	0	0.59	1.47	97.94

Decision Tree	AA	BA	CA	D	H	C
AA	97.39	2.03	0.29	0.29	0	0
BA	2.61	94.64	2.75	0	0	0
CA	0.61	3.65	95.74	0	0	0
D	0	0	0	95.59	1.91	2.50
H	0	0	0	1.91	95.29	2.80
C	0	0	0	1.47	3.24	95.29

SVM	AA	BA	CA	D	H	C
AA	91.16	7.39	1.45	0	0	0
BA	13.62	74.64	11.74	0	0	0
CA	4.57	16.29	79.14	0	0	0
D	0	0	0	78.35	3.83	17.82
H	0	0	0	16.64	75.26	8.10
C	0	0	0	14.43	6.19	79.38

Table 22. Results of ReliefF using 6-class idle and constant rpm accelerometer.

Naïve Bayes	AA	BA	CA	D	H	C
AA	84.06	9.57	6.37	0	0	0
BA	13.77	28.99	57.24	0	0	0
CA	5.94	7.91	86.15	0	0	0
D	0	0	0	82.62	5.45	11.93
H	0	0	0	24.89	64.80	10.31
C	0	0	0	39.32	40.06	20.62

KNN-3	AA	BA	CA	D	H	C
AA	96.96	1.59	1.45	0	0	0
BA	1.45	96.81	1.74	0	0	0
CA	0.76	3.35	95.89	0	0	0
D	0	0	0	97.79	0.74	1.47
H	0	0	0	4.71	89.10	6.19
C	0	0	0	1.47	3.39	95.14

Decision Tree	AA	BA	CA	D	H	C
AA	95.07	3.19	1.45	0.29	0	0
BA	2.17	92.32	5.51	0	0	0
CA	1.37	3.65	94.98	0	0	0
D	0	0	0	92.78	3.83	3.39
H	0	0	0	4.57	88.51	6.92
C	0	0	0	2.65	6.63	90.72

SVM	AA	BA	CA	D	H	C
AA	84.49	8.26	7.25	0	0	0
BA	16.23	58.41	25.36	0	0	0
CA	7	22.53	70.47	0	0	0
D	0	0	0	81.74	8.39	9.87
H	0	0	0	21.65	57.88	20.47
C	0	0	0	34.17	30.04	35.79

Table 23. Military vs. Civilian SFS classification results.

Naïve Bayes	M	C
M	100	0
C	0	100

KNN-3	M	C
M	100	0
C	0	100

Decision Trees	M	C
M	99.95	0.05
C	0	100

SVM	M	C
M	100	0
C	0	100

Table 24. Military vs. Civilian ReliefF classification results.

Naïve Bayes	M	C
M	100	0
C	0	100

KNN-3	M	C
M	100	0
C	0	100

Decision Trees	M	C
M	99.95	0.05
C	0	100

SVM	M	C
M	100	0
C	0	100